

Om et arbejde med computere

Af Preben Thomsen

Hvordan det begyndte

I det følgende har jeg nedfældet nogle af mine erindringer fra et lang liv sammen med computere. Hovedparten er skrevet kort efter min pensionering, men der er tilføjet en del senere efterhånden som det er dukket op i min erindring. Det har været meget spændende og interessant at være med i den kolossale udvikling der har været inden for elektronisk databehandling. I starten var maskinerne utrolig primitive - hvad vi heldigvis ikke vidste dengang - sammenlignet med de PC maskiner, man kan få i dag.

Medens det foregik, syntes vi at det var meget avancerede ting vi arbejdede med, naturligvis blev det meget mere avanceret i årenes løb, og jeg må da konstatere at EDB-verdenen slet ikke er gået i stå, fordi jeg har forladt den, så *meget* er altså sket efter dette er skrevet.

Efter en rimelig pæn studentereksamen fra Stenhus Kostskole i Holbæk og et par spildte år på Polyteknisk Lærestanstalt - nu hedder det vist Danmark Tekniske Universitet, blev jeg ansat i PensionsForsikringsAnstaltens beregnerkontor. Det var ikke så kedeligt, som det lyder. Faktisk er pensionsforsikringer vistnok de mest avancerede former for forsikringer - matematisk set. Jeg havde det fint der. Jeg kunne godt lide arbejdet, og det var bl.a. der, jeg traf min kone Kirsten.

Der var udfordringer, og jeg fik interesse for beregninger og talbehandling og vel også interesse for de mekaniske regnemaskiner vi brugte. Jeg fik også et '*Grundkursus for Hulkortoperatører*' på IBM-skolen. Jeg bestod den afsluttende test med maksimum point, hvilket *naturligvis* bevirkede at man flyttede min sidemand til hulkortafdelingen!!

PFA var et stift og konservativt foretagende. Man havde lønreguleringer én gang om året, og der kom så et år, hvor jeg *ikke* blev betænkt. Det var dengang, der var masser af jobs, så jeg begyndte at overveje at gå nye veje. Hulkortkurset havde givet mig interesse for databehandling, så jeg søgte et par job i den branche. Til mit held blev det IBM der ansatte mig som programmør-elev den 1. marts 1962. Og nej, jeg havde *ingen* anelse om hvad databehandling og computere var for noget, og de vage fornemmelser, jeg havde viste sig at være *helt* forkerte.

Jeg havde været glad for min tid i PFA, men jeg gik den tunge vej ind til min kontorchef for at sige op. Han så ikke glad ud, men bad mig gå ind til

personalechefen for at fortælle ham hvorfor jeg ville rejse. Jeg bad så personalechefen om en anbefaling, hvorefter han spurgte om mit navn!! Jeg havde været i PFA i 8 år, og der var kun ca. 120 medarbejdere, så jeg syntes nok det skulle være muligt for ham at kende mig.

Et par dage efter min ansættelse i IBM mødte jeg IBM's personalechef på gaden et stykke fra IBM. Jeg husker ikke hvordan, jeg havde mødt ham, men han lettede omgående på hatten og sagde: "*Goddag hr. Thomsen*". Han kendte mig altså godt nok, til at genkende mig *uden* for IBM, se sådan skal en personalechef være.

This page intentionally left blank.

Min første tid i IBM

Det der med elektronisk databehandling var noget helt nyt. Ikke bare for mig - jeg anede ikke en pind om, hvad det gik ud på og hvad jeg gik ind til - men også for de fleste i den IBM afdeling jeg var ansat i. På den tid benyttede man hovedsagelig hulkortmaskiner til databehandling. IBM var dengang stort set det eneste firma, der solgte kommercielle EDB maskiner, og ikke særlig mange, da de var *meget* dyre. I Danmark havde nogle kloge hoveder godt nok bygget DASK (det stod vistnok for **D**ansk **A**ritmetisk **S**ekvens **K**alkulator), der var den første danske elektroniske databehandlingsmaskine - bygget blandt andet af en hel del Nilfisk støvsugere. De blev vistnok brugt til køling fordi der var anvendt en umådelig mængde radiatorer, der udviklede en mængde varme. Men DASK var kun én maskine og altså ikke rigtig kommerciel!! Den var denne meget kreative gruppe bag DASK, der senere blev til det navnkundige RegneCentrale.

Da jeg blev ansat i IBM's Data Center Services (DCS) som programmør, fik jeg at vide at man snart fik en *stor* 1401 med en hukommelse på 8K. Jeg husker tydeligt at jeg ikke syntes at 8K (altså 8000) lød af særligt meget. Men sandheden var at et forsikringselskab havde anskaffet sådan en 1401 med kun 1,4K hukommelse. Denne maskine havde 6 magnetbåndstationer, og det var næsten umuligt at styre så mange båndstationer med så lille en hukommelse, båndstationer behøver rutiner, der bl.a. kan hjælpe ved læse og skrivefejl og sikre at man ikke skriver på bånd, der ikke må slettes - det var så svært, at man i Danmark havde været nødt til at udvikle et specielt sæt input/output makroer for at gøre det mulig (det har sikkert været en betingelse for købet). Sammenlignet med denne maskine var en 8K maskine altså ikke *helt* lille. Nu må man huske at hukommelsen i de dage bestod af en lille jernring for hver bit. Jeg mener at 1401 havde 12 bit (måske var der 13 fordi der også var en paritetsbit) i hver hukommelsesposition, altså næsten 100.000 små jernringe for at give 8000 positioner. Gennem hver af disse ringe var der trukket, vistnok, fire tynde tråde - alt dette var gjort i hånden, så det er let at forstå at hukommelse dengang var **meget** dyr og måske havde man derfor ikke sådan meget af det.

Jeg blev ansat som programmør. Nu er det at programmere ikke noget man normalt får lært af sin mor i forbindelse med den basale opdragelse, og jeg må da også erkende at jeg absolut ikke havde nogen som helst fornemmelse af, hvad jeg skulle til at lave. Jeg skulle naturligvis først uddannes. IBM havde en skole - den kendte jeg godt fordi jeg, som tidligere fortalt, havde deltaget i et grundkursus for hulkortoperatører. Det jeg nu skulle prøve at lære var naturligvis programmering i almindelighed og 1401 programmering i særdeleshed, men uheldigvis var der tre uger fra min ansættelse til sådan et kursus begyndte!! Tre uger!! Min chef besluttede at disse uger skulle tilbringes i DataCenteret's hulkortafdeling, så jeg kunne få en fornemmelse af, hvad et hulkort var, og måske også en fornemmelse af databehandling. Det lød ikke så slemt, men jeg blev *meget* klogere. Da DCS endnu ikke selv havde fået en 1401, kørte man opgaver om aftenen og om natten på en

maskine, der tilhørte et større forsikringsselskab. Der var så en IBM-operatør, der kørte på maskinen om natten - altid, næsten.

Jeg fik udleveret en kæmpe stak hulkort, mit gæt er at der nok har været omkring 10.000. Disse kort skulle forbehandles før de skulle køres på 1401. Jeg undrer mig stadig over hvorfor man brugte primitive hulkortmaskiner til at forberede noget til kørsel på en langt mere avanceret computer, men det skyldes nok at programmering var en yderst kompliceret og dermed meget dyr sag. Jeg husker ikke meget af, hvad der skulle gøres, men jeg husker bl.a. at der var to stammer af kort, der skulle blandes. Det havde man en maskine til - en såkaldt 'Collator'. En collator havde to steder man kunne stoppe kort ind, og vistnok fire lommer, hvor kortene kunne komme ud. Jeg husker det var lidt af en videnskab at programmere sådan en fyr. Hulkortmaskiner programmerede man ved hjælp af koblingstavler, en stor tavle med en masse huller. Man stak så ledninger ned i disse huller - masser af ledninger - ved hjælp af disse ledninger kunne man få maskinen til at løse den ønskede opgave, dette var en *slags* programmering. Programmeringen af min collatorkørsel voldte nu ikke problemer, den var så kompliceret at der var en fast tavle til kørslen - altså én, hvor alle ledningerne sad, hvor de skulle. Næh, problemet opstod under kørslen. Jeg husker stadig tydeligt at jeg omhyggelig fyldte kort i maskinen, der slugte dem med god appetit. Der gik rigtig lang tid før det gik op for mig at der *ingen* kort kom ud igen. Det er utroligt så mange kort, der kunne ligge inde i en collator, ligesom det også er utroligt så omhyggeligt sådan en collator kan folde kortene til smalle strimler - det lignede bælgen på en harmonika eller måske snarere en lampeskærm - og det var et sandt hestearbejde at få dem glattet ud så de kunne reproducere i en hullemaskine - ny mand i et nyt firma, det var *ikke* godt..

Senere skulle jeg sortere alle disse hulkort. En sorteringsmaskine var en ganske smart sag. Den er tilmed så enkel i sin konstruktion at man let kan forstå, hvordan den virker, og *så* er der noget at se på *hele* tiden - og det går *stærkt* - rigtig stærkt. I en kolonne i et hulkort kan der være huller 12 forskellige steder. Sorteringsmaskinen læser en kolonne af gangen og hælder kortene ud 12 forskellige lommer - afhængig af hvor hullerne er. Og det gik så hurtigt at man fornemmede at der dryssede hulkort i alle lommerne i en jævn strøm!! Idéen er så at man sorterer felterne bagfra, en kolonne af gangen så man på den måde opnåede den ønskede rækkefølge. Lommerne i sorteren kunne vel rumme omkring 1000 kort, så man kunne meget let risikere at man under en stor sortering var nødt til at tømme kortlommerne undervejs. Det skete for mig (naturligvis). Jeg tror at i en af kolonnerne var der hul samme sted i næsten alle kort. Resultat var at jeg havde stabler af hulkort på borde og i vindueskarme overalt i centret, 10.000 hulkort fylder en hel del. Det er muligt at en erfaren operatør kan klare sådan en situation - måske endda sandsynlig, det *var* jo på den måde man lavede databehandling i de tider. Men jeg var jo ikke erfaren, så jeg kunne ikke - slet ikke!!! Mindst én håndfuld kort blev lagt i en forkert bunke - midt i sorteringen. Det betød at den fejl der opstod blev spredt godt og grundigt blandt alle 10.000 hulkort.

Jeg var slet ikke populær morgenen efter man havde forsøgt at køre jobbet på 1401. Nat-operatøren var en meget sympatisk, venlig og tålmodig mand, som jeg senere fik stor fornøjelse af at arbejde sammen med, jeg er endog temmelig sikker på han helt har glemt denne episode. Han havde kæmpet en brav kamp for at rette fejlen/fejlene men han måtte opgive. Jeg husker ikke, hvor besværligt det var at rette fejlen - eller om jeg selv gjorde det, men jeg husker at jeg ikke syntes det var en speciel god måde at starte på et nyt job i et nyt firma. Jeg er nu nok den eneste, der stadig husker denne lille (?) episode (håber jeg). Men jeg husker tydeligt at jeg ikke syntes en sådan fiasko var en speciel god måde at begynde et nyt job.

På de tider var hulkort faktisk det foretrukne medie til dataopbevaring. Det vil sige der var et mægtigt lager af hulkort. De var placeret på store metalreoler. Hulkort-æskerne kunne rumme ca. 2000, og de stod side om side på højkant på hylder - naturligvis i et lokale midt i bygningen, et af dem der uden vinduer, og der var så folk der havde ansvaret for at holde styr på alle disse data, det var jo vigtige kundedata.

En dag skete det så!!! Der lød et øresønderrivende brag som ingen rigtig havde nogen forklaring på, jordskælv er jo ikke almindelige her i landet - men der blev jo på et tidspunkt behov for at hente en datafil fra hulkortlageret. Reolerne var væltet - dem alle sammen!!! Der lå hulkort hulter til bulter i *hele* rummet formodentlig op til ca. 1,5m højde. Det var totalt kaos, og jeg fatter stadig ikke hvordan det var muligt at få orden i alle disse kort igen, men det var heldigvis ikke mit problem!!!!

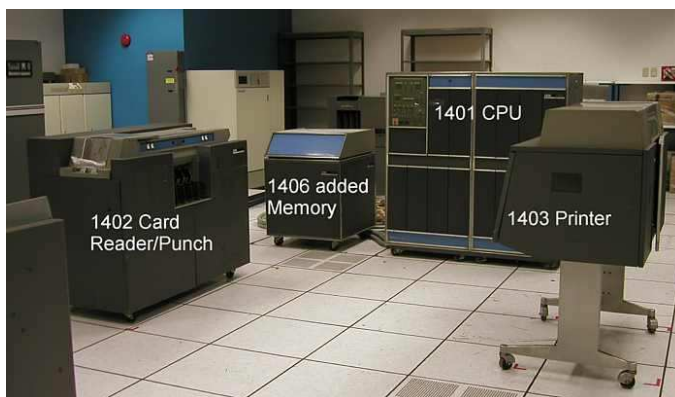
De tre uger gik dog, om end langsomt - jeg var slet ikke begejstret over at være 'fungerende hulkortoperatør' - men jeg kom da endelig på IBM-skolen, for at prøve at lære det jeg troede, jeg fremover skulle leve af. Dengang var jeg ikke klar over at jeg skulle blive ved med at lære *hele tiden*, resten af min tid i IBM - jeg er end ikke helt sikker på jeg er færdig endnu selv om jeg nu har været pensioneret et godt stykke tid, endelig er der jo meget andet end programmering i computer-verdenen.

1401 var den første kommercielle EDB maskine der blev populær. Det var en andengenerations maskine. Det betød at den var opbygget af transistorer i stedet for radorør. Det var en mærkværdig tingest. Den kunne leveres med op til 16K hukommelse - men en stor maskine (hvis der var mere end 4K) krævede naturligvis en ekstra boks, der stod for sig selv på gulvet. Den havde kun tre positioner til at beskrive adressen og tre positioner er ikke nok til at beskrive en adresse større end 999, hvis man benytter 10-tal systemet. Dette klarede man ved at tilføje ekstra bit's (2 stk.'s) til første og tredje adresse position. De tilsvarende bit's i midterste position blev brug til at adressere ét af tre index registre. Et index register blev evt. adderet til den pågældende adresse før denne blev brugt. Alt dette resulterede i adresser, der var næsten umulige at læse fra et storage dump (en udskrift af hele maskinens hukommelse), og en programmørs arbejde bestod for en stor del af at

læse dumps. Jeg må dog indrømme at disse index registre gjorde det muligt at lave ganske smarte programmer.

En anden af 1401's berygtede specialiteter var 'Word Marks'. Word Marks havde til opgave at afslutte felter, men de skulle også være i samme positioner som maskin instruktionerne. Det var de bestemt ikke altid!!! Et word-mark skulle sættes eller slettes med en kommando, heldigvis havde vi et primitivt oversætterprogram, der gjorde det meste af det automatisk.

1401 havde en hulkortlæser/huller, der kunne være op til 6 magnetbåndstationer og en lineskriver, der kunne skrive 600 linier/minut.



Jeg tror nok 1401 blev bygget som input/output maskine for større computere, sådan at forstå at man læste sine jobs fra hulkort til magnetbånd på 1401, derefter flyttede man båndet til den store maskine, der behandlede opgaverne og skrev resultater på magnetbånd, der så senere blev skrevet ud på papir af 1401. Fordelen ved denne teknik er at det er langt hurtigere at læse fra og skrive til magnetbånd end fra/til hulkort, man fik altså udnyttet den *store* maskines regnekraft langt bedre. Det er næsten den teknik, der senere blev kendt som SPOOL, men i ordet SPOOL ligger der i reglen, at det skal være on-line (SPOOL står for **S**imultaneous **P**eriferal **O**perations **O**n **L**ine). Det viste sig imidlertid at mindre firmaer ikke ville/kunne købe de store maskiner, så det blev ret almindeligt - i hvertfald i Danmark - at 1401 var *computeren* i et firma.

Programmering til 1401 var en rimelig kompliceret sag. Programmeringsproget hed autocoder. Autocoder havde mulighed for symbolske navne på instruktioner og makro facilitet (en makro er én enkelt instruktion, der udvikler sig til en hel række instruktioner - i regelen standard rutiner, f.eks. til input/output), men sproget var ellers meget tæt på programmering i maskinsprog.

1401-kurset bestod hovedsageligt i at lære om 1401's instruktioner (altså maskinens muligheder), men også at lære at bryde en opgave op i delelementer og beskrive

disse i blokdiagrammer. Jeg husker jeg var facineret. Det var noget helt andet end den Facit regnemaskine, jeg havde brugt i forsikringsselskabet, og langt mere spændende end at stable hulkort i vindueskarme, men der krævedes en omhyggelighed som jeg nok havde lidt svært ved at leve op til.

Efter kurset kom hverdagen. Min chef var en af disse effektive hurtigtænkere, så den første opgave fik jeg leveret som nogle notater på et stykke kladdepapir (sådan så det i hvertfald ud) og en viften med en finger i takt med en *meget* hurtig talestrøm. Jeg husker især at jeg var forvirret *meget* forvirret.

Jeg gik i gang. Når man havde besluttet sig til, hvordan man ville løse opgaven - altså tegnet et blokdiagram, begyndte man at kode. Det var en større sag, der krævede *ekstrem* omhyggelighed - ikke en af min stærkeste sider. Efter kodningen sendte man kodearkene til hullestuen, hvor venlige piger lavede de kodede instrukser om til hulkort. Nu havde man så sit program i form af en stak hulkort. Dette program kaldtes et sourceprogram - det var før Dansk Standardiseringsråd forvirrede os alle ved at opfinde danske navne for de begreber vi kendte så godt - sourceprogrammet blev senere - på dansk - kaldt et kildeprogram. For at gøre livet lidt lettere, var de hullede informationer som regel skrevet på kanten af kortene, så de var læselige, også for mennesker - dette er nu nok lidt overdrevet, vi kunne *alle* læse hullerne, det tog blot laaang tid for nogle af os.

Næste trin var at oversætte kildeprogrammet til noget 1401 kunne forstå. Det krævede en maskinoversættelse og maskinen var dyr, så dette skulle helst ikke gøres for tit. Resultatet af oversættelsen var et program bestående af maskininstruktioner, et objektprogram. Nu skulle programmet testes. Man læste programmet ind i maskinen sammen med nogle testdata, der gerne skulle simulere den virkelighed, programmet senere skulle arbejde under. Det gik hurtigt op for mig, at en computer gør nøjagtigt, hvad man beder den om - også selv om det er komplet tåbeligt. Man startede sit program - og lynhurtigt lyste en ækel rød lampe op - som tegn på at noget var galt. Derefter fik man en udskrift af maskinens hukommelse (et storage dump), og kunne så sidde og gruble over, hvad det var, der var sket. Når man så endelig havde fundet fejlen, kunne man gøre ét af to. Enten kunne man rette i maskinprogrammet, lave en patch - altså en rettelser i maskinprogrammet, eller man kunne rette i sit kildeprogram og lave en ny oversættelse. Da oversættelsen krævede maskintid, forsøgte man løse et antal problemer med patches. Det var imidlertid vigtigt hele tiden også at opdatere sit kildeprogram, så de patchedede rettelser også var med her. Glemte man det, kunne man lede efter samme fejl endnu en gang, efter næste program oversættelse, noget der var særdeles frustrerende, jeg ved det, jeg *har* prøvet det..

Jeg husker en mørk nat, hvor vi var to mand der kæmpede for at få et program til at makke ret. Der var en dum fejl, en division. Når en division kørte løbsk, kunne resultatet blive at en større del af hukommelsen, var blevet ændret hvilket gjorde fejlfinding næsten umulig. Vi havde omsider fundet fejlen, og lavet de nødvendige

patch-kort. Næste kørsel gav nøjagtig samme resultat. Det var *meget* mystisk, men vi brugte atter et par timer med at finde ud af at det var den samme division, der fejlede. Årsagen var naturligvis at vore patch-kort stadig lå på hullemaskinen, i stedet for i vort program dæk!!!

En patch blev typisk lavet på den måde at man erstattede en instruktion med en hopinstruktion (en '*branch*') til et 'tomt' sted i hukommelsen. Her kunne man tilføje en eller flere nye instruktioner og afslutte med at hoppe tilbage til den oprindeligt instruktionssekvens. Det hele blev naturligvis kodet i maskin-instruktioner i hulkort, det var lidt af en balancegang - de der 'wordmarks' skulle som tidligere fortalt stå under alle instruktioner, og det kom de ikke til af sig selv!!

Efter rigtig mange test's, hulning af utallige patch-kort og oversættelser lykkedes det som regel at få programmet til at gøre nogenlunde, som man havde tænkt sig. Det var imidlertid ikke altid det samme som det, 'kunden' havde tænkt sig. Når programmet kom til at køre med 'rigtige' data skete der ofte forbløffende ting. Til tider med det resultat at man måtte begynde helt forfra.

Der var sandelig gode grunde til at programmer var dyre!!!

Nu var det ikke alt sammen hårdt arbejde. Der var kreative mennesker, der havde lavet underholdende programmer. Tænk blot på hvad det har udviklet sig til med de moderne PC spil.

Der var et program, der hed EDIT eller var det EDITH?? EDIT lavede med bogstaver en primitiv tegning af en flot høj dame i lang selskabskjole - tegningen, der fyldte flere sider var forsynet med en tekst, der advarede mod at køre programmet igen med en switch sat (der var et antal kontakter, kaldet switches, som kunne aflæses af programmerne). Dette gjorde man naturligvis med det resultat at damen blev tegnet, kun iført trusser og BH, samt en kraftig advarsel mod at køre programmet endnu en gang med en anden switch sat. Dette skulle naturligvis prøves, hvorefter EDIT blev tegnet helt nøgen, men ærbart dækket af et skilt, der fortalte, at selv om 1401 var en yderst kraftig maskine, var der dog grænser for, hvad man kunne få den til. Jeg husker en søndag eftermiddag, da Imperial Biografen lukkede tilskuere ud på gaden. Vi havde besøg af en sælger i DataCenteret, der var placeret i stueetagen med store vinduer ud mod gaden, og han kørte EDIT for de forbipasserende. Der var ingen ende på jubelen, da det sidste billede blev vist. Jeg er temmelig sikker på der skal mere til at imponere nu om dage.

Andre programmer udnyttede at 1401 udsendte radiobølger under kørslerne. Dette kunne man udnytte ved at udføre bestemte sekvenser af instruktioner. Man kunne så placere en transistorradio oven på maskinen, og på den måde høre 1401 spille små melodier.

Der var dog en anden og mere mekanisk måde at få 1401 til at spille musik. Liniestriveren kunne skrive 600 linier i minuttet. Den skrev ved hjælp af en print-kæde, hvor bogstaverne var kædens led. Når et bogstav var i den rette position, slog en lille hammer bogstavet mod et farvebånd, der var foran papiret.

Når en linie blev skrevet, kom der en lyd, der varierede alt efter hvad der blev skrevet. Hvis man skrev de rette linier, kunne man tydelig høre at printeren spillede melodier, f.eks. *"I'll be coming round the Mountain, when I come"* (*"Du må få min sofacykel, når jeg dør"*).

Når vi slappede af, skrev vi forskellige pudsige programmer. Vi konkurrerede f.eks. om hvem der kunne skrive det mindste program (det *skulle* kunne lave noget nyttigt). Jeg vandt, jeg tror stadig jeg har programmet, der kunne være i ét hulkort (altså 80 positioner). Programmet kunne udskrive indholdet af hulkort på linieskriveren - der virkede faktisk. Hvorfor har jeg det??? Tja, hulkort har man ikke mere, linieskrivere virker nu om dage på en helt anden måde (der er vist i øvrigt ikke mere mange skrivere, der skriver linier) og det vil nok være svært at finde en 1401, der stadig kan køre, så det må vel nærmest være et sentimentalt kuriosum.

På trods af at vi havde det rart og hyggeligt i programmeringsafdelingen, tror jeg nok, jeg ret tidligt fandt ud af at det her programmering måske alligevel ikke rigtig var mig. Men heldet var endnu en gang med mig.

1620

Et forsikringsselskab havde købt en 1620 på betingelse af at IBM ville købe den overskydende tid tilbage. Det betød at man måtte have en 'Vor Mand på stedet'. Derfor var vi tre programmører, der fik en lyn-uddannelse i 1620. En systemkonsulent fra 'Matematisk Afdeling', der kendte 1620, brugte omkring en uge på at fortælle os tre programmører hvad 1620 var, hvorfor og delvis hvordan. 1620 var en lille matematikmaskine, det blev den i hvert fald kaldt. Vi fik naturligvis småopgaver, som vi skulle løse på maskinen. Jeg husker vi fandt et lille mellemrum mellem to bookninger i 1620-kalenderen, og dette kvarter var netop, hvad vi havde brug for til vor test. Det var samme mand, der havde lavet bookningerne før og efter *vor* tid, og han *havde* faktisk tænkt sig han de to bookninger skulle hænge sammen. Da vi dukkede op og viste ham bogen, måtte han jo nødtvunget indrømme at vi havde ret, men han så ikke glad ud slet ikke. Det viste sig heldigvis at han var umådelig flink - og dygtig også til 1620, så på trods at vi havde mast os ind i hans tid, gik han i gang med at hjælpe os, og hjælp havde vi sandelig brug for. Det kan selvfølgelig være af egoistiske grunde - han ville jo gerne have maskinen tilbage så hurtigt som muligt, men jeg kom senere til at kende ham vældig godt, og jeg er helt sikker på han *er* flink og meget hjælpsom, jeg var i øvrigt med til hans 25 års jubilæum meget senere, hvilket vel beviser at har ikke bar nag (i hvert fald ikke til mig).

Af disse tre programmører blev jeg (heldigvis) valgt til at blive 'eksporteret' til 1620. Jeg havde på det tidspunkt været i IBM i ca. et halvt år. Jeg er stadig imponeret over at det er muligt at få et fuldstændigt selvstændigt arbejde i et nyt firma så hurtigt.



Min opgave var at passe IBM's kunder på denne 1620, sådan at forstå at jeg skulle sørge for *alt*. Jeg skulle tage mig af tidsreservationer, sikre at der var papir og blanke hulkort og naturligvis de nødvendige oversætterprogrammer - og som noget helt nyt for mig applikations programmer. Vigtigst af alt skulle jeg være behjælpelig med løsning af programtekniske og andre problemer, kort sagt en slags alt

mulig-mand, egentlig et ganske stort krav til en nybegynder i branchen. Jeg ved ikke om det hørte med til mine opgaver, men jeg kom også til at yde støtte til ejerne af maskinen, ikke noget jeg havde noget imod. Hver gang jeg hjalp en eller anden, lærte jeg lidt selv. Lærdom kan man som bekendt ikke få for meget af, det var i hvert fald tilfældet for mig - især på det tidspunkt. Jeg fik sikkert også styrket min selvtillid, og *det* var der bestemt også behov for, jeg må tilstå jeg var lettere imponeret og ret bekymret over dette ansvar.

Jeg tror min chefs eneste interesse i 1620 var at han ikke ville have nogen som helst form for vrøvl fra 1620 fronten, måske især fordi 1620-kunderne ikke var hans kunder, de tilhørte matematisk afdeling. Jeg tror f.eks. slet ikke han besøgte mig en eneste gang på min nye adresse.

Dette var virkelig et kulturchock. Jeg var nyligt kommet fra et forsikringsselskab, hvor man stod pænt ved siden af kontorchefens skrivebord medens man fik instruktioner om hvad man skulle lave, han var i øvrigt en umådelig rar og venlig kontorchef. Nu var jeg pludselig min egen chef, jeg kunne stort set gøre hvad jeg ville, bare der ikke kom klager. I det år jeg arbejdede med 1620 i forsikringsselskabet så jeg kun min chef, når jeg på eget initiativ besøgte IBM. Lidt risikabelt var det dog at jeg meget hurtigt, måske for hurtigt - i hvert fald af kunderne - blev regnet som ekspert, en rolle man bestemt skal gøre sig fortjent til.

Min kontakt til IBM var gennem matematisk afdeling, og det var medarbejderne derfra, jeg kom til at arbejde sammen med, og det var deres sekretær, der blev min vigtigste forbindelse til IBM. Hun lød vældig sød, og jeg fik tit en hyggelig lille sludder med hende. En skønne dag fortalte hun at jeg skulle til møde med lederen af matematisk afdeling, men det var vistnok hende selv, der havde fået arrangeret mødet, hun syntes at vi nu havde talt så meget sammen at det var på tide vi fik set hinanden. Vi vedblev at være gode venner gennem rigtig mange år, også længe efter vi ikke havde nogen arbejdsmæssig forbindelse, hun var da også med til mit 25-års jubilæum og jeg var med til hendes.

1620 kunderne var kunder fra 'Matematisk Afdeling'. De var alle umådelig flinke, de fleste var ingeniører, og mine kollegaer fra Matematisk Afdeling var yderst hjælpsomme, så på en eller anden måde gik det. Det gik godt. Ja faktisk *rigtig* godt. Jeg opnåede da at blive anset for 1620 specialisten, ikke *en* 1620-specialist, men *specialisten*. Det blev så min første specialist rolle, men heldigvis kom der en hel del specialist roller til mig senere.

1620 var en anden generations computer ligesom 1401, hvilken betød at den var opbygget af transistorer, og ikke som førstegenerationsmaskinerne af radorør. 1620 blev anset for at være en matematikmaskine. Det var der flere grunde til. Der var dels dens *store* hukommelse - den havde mindst 20K og maksimalt 100K altså en hel

del mere end 1401, og så kunne man få en FORTRAN oversætter til den. FORTRAN er et High-Level Language - altså et sprog, der mere tager hensyn til programmøren end til computeren. FORTRAN var vistnok det første sprog af denne type, og da det stadig anvendes, må man sige at det har haft en *vis* succes. På 1620 var alternativet til Fortranprogrammering et mere maskinsprogagtigt sprog, dette sprog kaldtes SPS (Symbolic Programming System). SPS var om muligt endnu mere primitiv end 1401's autocoder, der var f.eks. ingen mulighed for makroer.

Egentlig var 1620 ikke særlig matematisk, den kunne f.eks. kun regne ved hjælp af tabeller, der blev indlæst i dens hukommelse sammen med programmerne. Der var én tabel til addition og én til multiplikation, subtraktion og division er som bekendt specialtilfælde af addition og subtraktion - disse tabeller kunne naturligvis ændres (de lå jo i maskinens almindelige hukommelse), med det resultat at 1620's regne-regler blev anderledes. Instruktionssættet i 1620 var mindre og mere primitivt end i 1401. Instruktionerne have en fast længde, og der var 5 positioner til adresserne, så de kunne umiddelbart læses, hvilket ville have været en stor fordel, hvis man havde kunnet få et storage-dump - man ak, denne 1620 havde *ingen* hurtigskriver. Felterne i 1620 var af variabel længde. Man brugte et 'flag' som feltadskiller, det var næsten som 1401's wordmark, Der skulle vistnok også være et flag over maskininstruktionerne.

1620 kunne leveres med papirbånds læser/huller og/eller hulkort læser/huller. Papirbånds-udstyret til 1620 var temmelig håbløst i forhold til datidens standard, så heldigvis havde 'min' maskine hulkort, det var nok også fordi IBM gik kraftigt ind for hulkort. Der var en skrivemaskine som konsol - en ganske almindelig IBM skrivemaskine, modellen fra før 'golfkuglen' blev opfundet. Dette var den eneste skrivemulighed for 1620 og altså så langt fra en hurtigskriver som man kunne komme - måske bortset fra en gåsefjer.

Oversættelsen af et SPS program kunne tage adskillige timer, hvis man ønskede en program-liste udskrevet, og en programliste var næsten uundværlig, når man skulle fejlsøge i programmet. Heldigvis havde forsikringsselskabet en 407 tabulator (en hulkortmaskine, der bl.a. kunne skrive med mere rimelig hastighed), og det var muligt at få SPS oversætteren til at producere programlisten i form af hulkort - to kort pr. linie. Ved hjælp af en speciel snedig koblingstavle til 407, var det muligt at udskrive listen på 407 meget hurtigere end skrivemaskinen kunne klare det. Koblingstavlen var heldigvis beskrevet i 1620 litteraturen, så vi fik fremstillet sådan en tavle, så vi med held kunne anvende denne teknik, og det virkede faktisk

Min kære schweizerkniv vandt en dag en mindre sejr over en tekniker. Teknikeren skulle montere en ny/større motor i vores kortlæser, så den kunne køre dobbelt så hurtigt. For at fjerne den gamle motor, måtte han save et par metalstivere over. Han mumlede surt noget om at han nu var nødt til at gå op på værkstedet for at finde en nedstryger. Så var det jeg foreslog ham at låne min lommekniv. Han var lidt

nedladende, han påstod at han behøvede *rigtigt* værktøj. Jeg fortalte ham så - vistnok også lidt nedladende - at han da kunne give kniven en chance. Da han så hvor god saven i kniven var, kunne han sagtens indse at det ville være dumt at spille tid med at fremskaffe en 'rigtig' nedstryger. Jeg ved ikke hvad han syntes om disse udmærkede knive, men han holdt da op med at være nedladende overfor min.

1620 blev min første store computerkærlighed. Det var en pragtfuld maskine at arbejde med, og jeg *ved* at alle, der havde med den at gøre, syntes som jeg. Jeg har stadig et farvebillede af den hængende - og jeg ved helt bestemt, hvor jeg skal søge, hvis det en dag skulle forsvinde på mystisk vis! Jeg ved egentlig ikke hvorfor jeg var så glad for 1620, den var mere primitiv end 1401, den var langsommere og så manglede den desperat en linie-skriver. Til gengæld var den billig. Det betød at der var mindre stress over arbejdet ved den. Man kunne tillade sig at undersøge hvordan den arbejdede, altså *virkelig* gå i dybden. Det var morsomt, og i mit job var det absolut nødvendigt - jeg skulle jo være specialisten så man forventede jeg kunne svare på alle spørgsmål, men heldigvis havde jeg tiden at studere maskinen grundigt og heldigvis havde maskinen ikke alt for travlt. Jeg fandt altså efterhånden ud af hvordan en computer arbejder, og det temmelig detaljeret. Jeg har senere arbejdet med mange andre computere, nogle der arbejdede *helt* anderledes, men det jeg lærte på 1620 har jeg hele tiden haft stor glæde af. En anden forskel på 1620 og 1401 var arbejdet, der blev udført. 1401 blev brugt til at køre administrative opgaver for afdelingens kunder, og administrative opgaver er altid noget meget højtideligt, og noget der (helst) skal passe på øre. På 1620 blev der kørt alle mulige mærkværdige opgaver, mest for ingeniører, og disse opgaver var ofte af en natur, hvor man ikke kunne være helt sikker på resultatet, så det var altså ikke helt så alvorligt hvis noget ind imellem gik mindre godt. Den slags synes jeg er langt mere morsomt og det gjorde helt bestemt arbejdet langt mindre stresset.

Det uundgåelige måtte jo ske, telefonen ringede og den første kunde bestilte tid. Han var så venlig at fortælle mig, hvad han havde tænkt sig at bruge maskinen til, så jeg fik mulighed for at læse på lektien. Det gav mig en chance for at give ham en fornuftig vejledning. Det gik. Jeg ved ikke om det var fordi han var høflig eller venlig, men han gav *ikke* udtryk for at han havde bemærket, hvor lidt jeg vidste om 1620 på det tidspunkt.

Den første tid blev virkelig en vanskelig vandring på en *meget* smal sti. Når en kunde meldte sin ankomst fik jeg meget travlt med at forberede mig, og jeg var ofte kun ganske få skridt foran kunderne - til tider kun ca. en halv time. Til mit held var der ofte pauser, hvor 1620 var ledig. Denne tid benyttede jeg til at udforske maskinen, og jeg opnåede virkelig - efterhånden - at forstå de fleste af 1620's funktioner. Jeg endte faktisk med at blive rigtig specialist - og det var ikke bare efter min egen opfattelse!!!

Dette at have lært en datamaskine at kende næsten til bunds, har senere ofte gavnet mig, når jeg er blevet præsenteret for mærkværdige problemer. Andre maskiner virker slet ikke på samme måde, men alligevel

Jeg havde det ganske pragtfuldt sammen med 1620. Kunderne kom og gik. På den tid var forventningerne ikke så høje, som de er nu, så hvis det gik godt var man *meget* glad. Gik det mindre godt, tog man det med godt humør, man kunne jo bare prøve igen. Det var en rigtig pioner-tid, vi prøvede alle at forstå hvad der foregik og at blive dygtigere. Vi forstod det ikke altid, men jeg tror nok vi blev dygtigere, hvis ikke, havde der nok ikke været så mange computere i dag.

Hvad brugte man så maskinen til?? Det er jo længe siden, så jeg husker ikke alt, men jeg husker en ingeniør, der var interesseret i statistik. Han kom ofte - når han ikke var på skadestuen med ét af sine mange børn - og kørte statistiske beregninger.

En ingeniør fra et større rederi forsøgte ved simulering at beregne hvornår tankene i tankskibe var så tærede at de skulle skiftes ud. Der ville kunne spares masser af penge, hvis man kunne undgå at sende skibene på værft og adskille tankene for at måle på dem.

Et foderstof-firma fra Århus beregnede hvordan kyllingefoder skulle blandes. Der er visse krav til sådan noget foder, det skal mindst indeholde en givet mængde af forskellige vitaminer og mineraler. Man blander foderet af forskellige ingredienser, der indeholder disse ting. Ingrediensernes priser varierer, så opgaven var at blande foder, der opfyldte alle kravene, og som var så billigt som muligt - ikke nogen helt enkelt beregning. Det var åbenbart meget vigtige kørsler, for jeg husker at jeg til tider - ofte sene aftener - måtte et smut ned på hovedbanegården, for at sende resultaterne som ekspresbrev - andre gange måtte jeg ringe til Århus og læse resultaterne i telefonen fordi det *virkelig* hastede. Tænk hvor meget lettere alting er nu om dage med elektronisk post. En dag bestilte han tid, han ville komme personligt. Det viste sig at han syntes at vi havde arbejdet så meget og godt sammen, at han nu ville se 'giraffen'. Forhåbentlig blev han ikke alt for skuffet.

Et elsselskab beregnede elektriske netværk. Det var noget med at der skulle være strøm overalt i nettet, også selv om nogle transformatorerne brød sammen og heldigvis var der et standard program til 1620, der kunne beregne sådan noget.

Og så var der undervisning. IBM skolen havde et FORTRAN kursus, der blev afsluttet med løsning af en praktisk opgave. Kursisterne skulle altså skrive et program og derpå og prøve at få det til at virke. Da 1620 på det tidspunkt var den eneste maskine der kunne FORTRAN, var der altså hands-on på 1620. Det gik normalt fint og efter planen, men der var en enkelt gang, hvor det var lidt anderledes. Der kom pludselig en opringning til instruktøren, han blev synligt forvirret, fik så fat i mig og spurgte om ikke jeg kunne afslutte kurset for ham., hans kone var nemlig ved at føde!! Jeg ville naturligvis gerne hjælpe ham, men på det tidspunkt

havde jeg ikke selv været på FORTRAN kursus, så jeg havde kun en meget vag fornemmelse af, hvad FORTRAN var. Naturligvis gik det, kunderne var jo flinke, og jeg var blevet temmelig dygtig til at improvisere. Jeg har vist i øvrigt stadig ikke været på FORTRAN kursus, men jeg har da skrevet et par brugbare FORTRAN programmer.

Der var en herlig stemning under disse kørsler. EDB var helt nyt for os alle. Vi stod spændte og stirrede på alle de flotte lamper der glimtede lystigt (dengang lignede maskinerne lidt bedre vittighedstegnernes computere), indtil alt pludselig stod stille. Så var det *jeg* skulle forsøge at finde en plausibel forklaring, eller endnu bedre, pille lidt, så programmet kørte videre eller evt. komme med et forslag til, hvordan vi kunne få det til at gå bedre næste gang. Så vidt jeg husker var det meget almindeligt at man skulle prøve adskillige gange. Et af de større problemer var nok at programmerne ikke checkede input data tilstrækkelig. Når man har prøvet at programmere, lærer man hurtigt at det er forholdsvis enkelt (meget lettere) at lave et program, der kører rigtigt, hvis det kun får korrekte data. Det er langt mere besværligt når programmet skal sikre sig at data er i orden. Tænk bare på hvor svært det er at checke om en dato er korrekt - rigtig antal dage i måneden, skudår og alt det der. Jeg tror ikke gamle Gregorius tænkte på computere, da han i sin tid designede vor kalender.

Jeg kan huske at jeg til tider rettede programmer i maskinens hukommelse ved hjælp af skrivemaskinen, og når de så kørte, kunne jeg overføre programmet fra hukommelsen til hulkort som et nyt program, så man på den måde endte op med et køreklart program, noget man *bestemt* ikke ville kunne gøre med dagens maskiner.

Det var i denne periode jeg fik skrevet mit første FORTRAN program. Det var noget med reducerede priser til landmænd, hvis der havde været urenheder i den mælk, de havde leveret til mejeriet. Jeg tror nok det hed 'Reduktasefradrag'. Problemet var at dette fradrag blev udregnet efter en logaritmisk skala. Dengang var logaritmer stadig noget med store støvede tabeller, altså noget besværligt og dermed noget med store fejlmuligheder. FORTRAN er et matematisk sprog, så det har naturligvis også en logaritmefunktion, så det var en lille let sag at lave denne, før så indviklede beregning.

Det var ikke meget forsikringsselskabet brugte maskinen, men der var en aktuar der blev interesseret i den. Og da jeg nu var der, kunne jeg lige så godt støtte også ham, det var selvfølgelig (måske) ikke mit job, men måske kunne jeg også lære ved at hjælpe ham og han var jo IBM-kunde, og endelig syntes jeg jo det var sjovt at få ting til at virke. Han kom på et Fortrankursus, der blev afsluttet med at man skrev et program. Hans program kørte bare! Efter nogle dage havde han skrevet et nyt program. Det blev oversat, han kørte det, fik sine resultater, og gik ud af døren som om det var den naturligste ting af verden. Da han lavede sit tredje program, lærte

han at træerne *ikke* vokser ind i himlen - slet ikke. Nu gik det ham, som det plejede at gå alle os andre. Det virkede ikke, han var tæt på at bryde sammen - det plejede jo altid at gå så glat, men heldigvis lykkedes det mig hurtigt at finde den lille banale fejl, han havde i sit program, og så var han glad igen. Dog undrede han sig over at jeg så hurtigt kunne finde fejl i *hans* program: 'De ved jo ikke en gang, hvad programmet laver!' Han var ikke EDB-mand, så derfor sagde han 'De'. Når jeg så hurtigt kunne hjælpe ham, var det måske fordi jeg omsider var ved at blive *rigtig* specialist, jeg havde jo efter hånden fået en del erfaring i at løse den slags problemer.

Jeg husker et andet morsomt problem, han løb ind i. Han havde skrevet et program, hvor der var en sammenligning af nogle data. Han kunne ikke forstå at programmet påstod at data var ens selv om både han og jeg tydelig kunne se, de var forskellige. Det var et af de problemer, det ville have været vanskeligt at løse uden et indgående kendskab til maskinen. Problemet var at han var kommet til at nulstille maskinens aritmetiske tabeller, og maskinen foretog en sammenligning ved at subtrahere de to tal, der skulle sammenlignes. Hvis resultatet blev nul, *var* de i følge maskinen, ens. Med nulstillede tabeller ville *alle* udregninger og sammenligninger give resultatet nul!! *Derfor!!!!*

Tre ærværdige ingeniører dukkede op en dag. Vi havde et 1620 program, der kunne lave netværksstyring - altså noget med projekt planlægning. Sådan noget med at en murer ikke kunne begynde at stable sten, før grunden var støbt og tømreren ikke kunne rejse taget før mureren blev færdig. Disse ingeniører, der kom fra en større virksomhed, havde et mægtigt projekt, som de mente, de havde gennemarbejdet ret grundigt. De bad så 1620 om at prøve det samme, og efter en times tid (eller var det mere??) nedkom 1620 med *sit* resultat. Der rejste sig et latterbrøl hos de tre herrer, da det gik op for dem at deres projekt rent faktisk kunne klares tre uger hurtigere end de selv havde beregnet.

Det var ikke alt på 1620 der var direkte relateret til mit arbejde. Jeg tror det var handelshøjskolen, der skulle køre en konkurrence om noget virksomhedsstyring, og det skulle naturligvis foregå på 1620. Da der var brug for mange hænder, blev jeg spurgt om jeg kunne tænke mig at deltage. Naturligvis svarede jeg positivt, jeg var jo interesseret i at lære så meget som muligt. Det var nu ikke meget jeg fik lært ved den lejlighed, mit arbejde bestod i at samle hulkort sammen og putte dem i 1620. Ikke særlig interessant, men på en eller anden måde var det par hyggelige aftener, jeg brugte ved den lejlighed. Da vi var færdige blev vi inviteret ind på "Syv Små Hjem", hvor vi fik rejemadder (af den rigtige fornemme slags) og hvidvin som tak for vor indsats. Ydermere blev jeg et par dage senere ringet op og fik besked på at kigge forbi, da der stod en pose med et par flasker vin til afhentning. Den slags var vistnok ikke helt i IBM's ånd, man måtte jo ikke få gaver af kunderne men da jeg ikke sladrede, skete der selvfølgelig ingen ting.

En dag blev jeg ringet op af en ingeniør. Han fortalte at han var temmelig interesseret i disse nymodens computere, om jeg havde noget materiale, han kunne få, så han kunne lære lidt om dem. Jeg gav ham en reference manual, og troede fuldt og fast at den ville skræmme ham langt væk, sådan én er *meget* teknisk. Men nej!! Nogle dage (uger??) senere ringede han og bestilte tid. Han dukkede op med en stak hulkort under armen. Disse kort puttede han i maskinen, og da en masse lamper viste at det var *helt* galt, bad han mig om hjælp til at få vist nogle positioner i 1620's hukommelse. Dernæst sagde han 'Åh ja' og gik bort med et forklaret udtryk i ansigtet. Et par dage senere dukkede han op igen, og nu kørte hans program!! Så *måtte* jeg have en forklaring. Han havde skrevet sit program i maskinsprog, han havde selv skrevet en load-rutine, han havde nulstillet 1620's hukommelse og han havde indlæst aritmetiske tabeller - alt hvad man skal gøre for et 1620-program kan køre. Han havde i sandhed læst på lektien og gjort sit hjemmearbejde, og gjort det godt. Aldrig før eller siden har jeg oplevet sådan noget. Hans problem - det eneste der var, var at han havde nulstillet maskinen efter han havde indlæst de aritmetiske tabeller. Han var oven i købet flov over at have dummet sig. Jeg tror ikke jeg så ham siden, men måske fik han adgang til en anden maskine, i hvert fald synes jeg det ville være synd, hvis en mand med hans evner for computere ikke udnyttede dem.

Jeg ved ikke om det var ham der var årsagen, men jeg blev selv kastet ud i et projekt, der krævede kodning i maskinsprog. En af de flinke kollegaer fra matematisk afdeling var meget interesseret i statistik - han er egentlig kemiker, men altså . . . En af de faciliteter han savnede i 1620-FORTRAN var mulighed for at danne tilfældige tal, og da jeg havde god tid indvilligede jeg. Jeg fik en metode og gik i gang. Nu er det der med maskinsprogs programmering ikke helt simpelt, og rutinen skulle integreres i FORTRAN oversætteren. Hvordan gør man det?? Det fandt jeg en beskrivelse af, men maskinsprog!!! Hvis man har en hop instruktion, skal man rette adresser hver gang man tilføjer en instruktion mellem hop-instruktionen og destinationen. Man lærer i sandhed at sætte pris på oversætter programmer, selv de mest primitive.

Jeg mener at huske at randomiseringsrutinen fyldte ca. 500 karakterer, den var rimelig kompakt og opfyldte alle de krav, der blev stillet til den f.eks. var det mulig at gentage de tilfældige tal, hvad statistikere af en eller anden grund har brug for. Vi skrev et lille testprogram, der lavede tilfældige tal i timevis (dagevis??) og samtidig checkede at tallene virkelig var tilfældige. Jeg husker ikke hvordan, men det var noget med at alle tallene skulle forkomme lige hyppigt - næsten, men de måtte *endelig* ikke komme for regelmæssigt. Hvad jeg ikke ved er om rutinen nogensinde blev brugt, men det gør ikke så meget, det var sjovt og lærerigt at skrive den og vistnok det eneste større jeg nogensinde har lavet i maskinsprog , samtidig må jeg må da erkende at jeg mange gange senere også har lavet rimelig

smarte og ret komplicerede ting, der *aldrig* er blevet brugt. Sjovt nok har det aldrig gjort mig noget. Hvis jeg har lavet en ting, og føler med mig selv at den er lavet godt, er jeg tilfreds med *det*, men jeg er naturligvis endnu mere glad, hvis det bliver brugt og gør gavn, men det er vist en helt anden historie. Til gengæld fik denne lille opgave lært mig at det rent faktisk er muligt at modificere og udbygge standard programmer, noget jeg senere har gjort særdeles meget.

Jeg fik mig også rodet ind i et lille sjov projekt, der ikke havde noget med IBM at gøre. En af mine venner fra min PFA tid fortalte over en stille øl (eller var det mon to??) at PFA havde købt en brugt 1620. Det vidste jeg naturligvis godt, jeg tror de gav 585.000 kr. for den, men hvad jeg ikke vidste var at man ikke rigtig brugte den til noget fornuftigt. Den blev bl.a. anvendt til at interpolere tabeller med tariffer, så man kunne bruge disse i en hulkortkalkulator. Jeg fortalte naturligvis forarget at det var det rene pip da 1620 sagtens kunne interpolere tabeller og lave hulkortkalkulatorens arbejde i én arbejdsgang, hvorved hele processen ville blive meget enklere og sikkert også meget hurtigere. Vi besluttede os til at bevise det, så vi valgte forsikringerne fra et af de mest besværlige firmaer (altså set fra beregnerens synsvinkel), det alle i beregnerkontoret hadede at arbejde med. Vi gik i gang med at skrive et program, der kunne håndtere disse beregninger. Kodningen blev skrevet på almindeligt skrivepapir, og min tidligere kollega fik papirene omsat til hulkort på PFA's hullestue. Jeg kom aldrig i nærheden af maskinen, oversættelse og aftesting blev foretaget på afstand. Jeg forklarede meget grundigt, hvad der skulle gøres og hvad jeg skulle bruge, hvis/når noget gik galt. Det tog sikkert lang tid og mange forsøg, jeg husker det ikke, men det lykkedes faktisk at få programmet til at køre og køre rigtigt, så det burde egentlig være en god demonstration af 1620's muligheder over for PFA. Jeg husker ikke jeg nogen sinde fik at vide om man blev glad for programmet, måske var det et problem for EDB afdelingen, da det jo demonstrerede at *de* ikke havde set alle mulighederne ved 1620 - jeg ved det ikke, men det var da sjovt at lave det, og vel ekstra tilfredsstillende at det kunne lade sig gøre under sådanne vanskelige betingelser, men der gled vel også et par øller ned under processen - de burde have været betalt af PFA!!!! Program testning er nu en gang lettere, når man kan stå ved siden af maskinen og kan følge med i, hvad der foregår.

En dag var der et problem med konsolskrivemaskinen. Tasten der indsatte flag-karakteren virkede ikke som den skulle. Når en normal karakter var indtastet, skulle adressen forøges med én, så den var klar til en karakter i næste hukommelsesposition. Dette var ikke tilfældet med flag-karakteren, den skulle sidde i samme position som en anden karakter (på samme måde som en accent på en skrivemaskine). Men der var en kontakt, der sommetider blev snavset, og så virkede det ikke, som det skulle. Det var ikke noget stort problem, jeg ringede til min udmærkede tekniker, som jeg havde det særdeles fint med. Han kom, skilte skrivemaskinen ad og rensede den snavsede kontakt. Før han samlede det hele igen,

ville han lige prøve om det virkede. Det gjorde det ikke. Så snart man rørte en tast, en hvilken som helst tast, tændte samtlige lamper på maskinens konsol, og der var mange - ligegyldigt hvilken tast, *alle* lamper blev tændt. Det så rigtig flot ud, men den stakkels tekniker blev ligbleg. Jeg tror aldrig før jeg havde set nogen så tæt på et sammenbrud, men det *var* naturligvis i starten af min tid i IBM, senere har jeg mange gange set tilløb til større nervesammenbrud.

Jeg ordinerede omgående en kop kaffe med blødt brød på den anden side af Kongens Nytorv. Når tingene ser *så* slemme ud, skal de fortrænges, den slags skal på afstand - omgående!!! Jeg har mange gange oplevet at problemer bedst løses ved at få dem på afstand, og ganske rigtig, da vi kom tilbage kunne jeg med det samme se, hvad problemet var, jeg tror teknikeren stadig var i chok. Skrivemaskinen var uden fødder stillet på bordet, så den hvilede på kontakterne, der således alle var sluttede. Derfor .

Alt godt får en ende. Mit arbejde med 1620 sluttede, da forsikringsselskabet solgte maskinen i øvrigt til PFA. Det betød at jeg måtte vende tilbage til folden og til 1401 programmeringen, men min tid på Kongens. Nytorv tænker jeg stadig tilbage på, som en af de bedste perioder, jeg har haft i IBM, måske fordi jeg for første gang prøvede at arbejde helt selvstændigt, og også fordi jeg fik lejlighed til at lære en maskine at kende - helt til bunds.

Da NEUCC (Nordic European University Computing Center) blev oprettet i forbindelse med en gave fra IBM, fik jeg en ganske fornøjelig opgave. IBM's gave var en model 7090. Det var en meget stor binær 'talknuser'. Den var så stor at den benyttede en 1401 som input/output maskine. Naturligvis kom den lille maskine først, så man stod med et center uden det vigtigste. Heldigvis var der kommet en FORTRAN oversætter til 1401, og da jeg kendte lidt (meget lidt) til matematiske kørsler og FORTRAN fra mit 1620 engagement og samtidig havde kendskab til 1401, blev det mig der fik fornøjelsen af at få oversætteren installeret og afprøvet. Jeg drog ud til Lundtoftesletten, installerede FORTRAN oversætteren, d.v.s. den blev anbragt på et magnetbånd. Den *kunne* faktisk køre, og da den hele tiden skrev på printeren under oversættelsen, fik man indtryk af at det gik gevaldigt hurtigt. Når bare man ser der sker et eller andet hele tiden, er man ikke nær så utålmodig.

Senere - meget senere - havde jeg en snak med den IBM'er, der var med til oprettelsen af NEUCC, og som arbejdede som konsulent på stedet. For sjov spurgte jeg ham hvordan han ville rangere 7090 ind i den moderne produktlinie. Han tænkte sig om, og pegede så på en gammel PC/XT og sagde at det ikke kunne blive til mere end den der. Da NEUCC blev oprettet, skrev pressen at denne nye maskine var så kraftig at den fordoblede Danmarks samlede datakraft. Joh, det gik hurtigt, også den gang.

Fordi Data Centeret ikke mere havde adgang til 1620, fik jeg endnu en opgave med 1401 og FORTRAN. Vistnok en af mine sidste 1401 opgaver. Et større bilfirma havde behov for nye tabeller over ydelser ved afbetaling - problemet var naturligvis at rentesatsen havde ændret sig. Rentesregning er ikke helt enkelt, så FORTRAN var lige sagen. Jeg husker jeg fik udleveret at stak papirer. Det var beregningerne, som en aktuar havde foretaget, da tabellerne sidst blev beregnet. Dette mindede lidt om mit arbejde i forsikringsselskabet, så jeg var næsten på hjemmebane. Programmet blev meget enkelt. Et par programsøjfer, renteformlen og resultaterne skrevet pænt ud på papir. Et stykke kage som amerikanerne siger (a piece of cake). Da programmet kørte, blev det naturligvis først kørt med den 'gamle' rentesats, for at checke det mod aktuarens beregninger. De første par tal passede, men så begyndte der at komme små unøjagtigheder. Det måtte naturligvis undersøges, så jeg fandt en regnemaskine - magen til den jeg kendte så godt fra min forsikringstid. Jeg brugte en hel aften på at gennemgå aktuarens beregninger. Efterhånden fandt jeg ud af at han havde beregnet det første tal 'rigtigt' - det stemte med 1401. Han havde derefter brugt dette tal som udgangspunkt for beregningen af næste tal o.s.v. naturligvis for at slippe lidt lettere fra alle disse besværlige beregninger, det var jo længe før lommeregneren blev opfundet, så det eneste regnemaskinerne kunne var addition og multiplikation. Det betød at afrundingsfejlen fra det første tal efterhånden blev større og større, hvorfor der blev en afvigelse fra 1401's tal, der alle blev beregnet så nøjagtigt som muligt - nemlig direkte ved hjælp af renteformlen. Det var en ganske god fornemmelse af fortælle bilforhandleren, hvordan *hans* tabel var lavet, og at tilbyde at vi sagtens kunne lave den nye tabel på samme måde, men at vi naturligvis ville anbefale at gøre det 'rigtigt'. Det blev naturligvis de rigtige tal der kom til at stå i nye tabeller. Jeg husker jeg var vældig stolt af at kunne producere noget der var mere korrekt end en aktuar's beregninger. Husk på, jeg havde stor respekt for aktuarer fra min forsikringstid.

Som programmør fik man mange meget forskelligartede opgaver. Der var f.eks. et nyt bilforsikringsselskab, der var anderledes fordi de forsikrede blev delt op i grupper, der havde en fælles bonusordning. Idéen var at f.eks. medarbejdere fra et firma skulle i samme gruppe - de kendte jo hinanden!!! Sådan en sag er ikke helt enkel at administrere, så der skulle et EDB program til. Jeg fik fornøjelsen - måske fordi jeg havde en fortid i et forsikringsselskab??? Det blev en morsom og ret stor opgave, som jeg havde ret megen fornøjelse af, også fordi jeg havde et fortrinligt samarbejde med kundens kontakthand.

Sammen med en seniorprogrammør fik jeg fik en anden spændende, men mindre vellykket 1401 opgave. Da han var senior, var jeg nærmest hans 'føl', på den tid var mit speciale 1620. Vi fik opgaven defineret. Det var et bogholderi og lagerregnskab for en industrivirksomhed, der lavede papdåser og ølkanister. Jeg anede på det tidspunkt *intet* om regnskab og bogføring, det gør jeg vist stadig ikke. Men min

'senior' så fortrøstningsfuld ud, så jeg stolede på at *han* vidste, hvad han gjorde. Hvor naiv kan man være???

Vi gik i gang med kodningen. Der skulle skrive en række programmer. Data var naturligvis på magnetbånd, hvad der måske var en væsentlig årsag til vore problemer.

Vi fik kodet nogle programmer, og vi fik nogle testdata fra kunden. Uheldigvis svarede vort output langt fra til kundens forventninger. Så gik vi i gang med fejlsøgning lavede nogle rettelser og prøvede igen. Det blev anderledes, men stadig ikke lige det, der var ønsket. Sådan gik der en rum tid. I de dage var en rum tid virkelig noget. Masser af lange nætter, hvor vi sad med næsen begravet i storage dumps, på jagt efter fejlplacerede wordmarks og divisioner, der løb løbsk.

Min 'senior' besluttede sig til at søge nye græsgange altså et job i et andet firma. Det var måske fornuftigt af ham, men jeg stod altså alene med disse programmer, der ikke ville makke ret.

Så var der en aften, hvor vi havde kørt en test - det var sammen med kundens bogholder - en meget dygtig, venlig og tålmodig mand. Det gik stadig galt. Så inviterede bogholderen mig på en middag på Hotel Royal's cafeteria - de var berømte for deres tournedos med bearnaisesauce. Da vi havde spist, tog han en serviet, og begyndte at tegne og fortælle, han forklarede mig tålmodigt, hvad bogholderi gik ud på, et mini bogholderikursus. Det der med at kredit var det, der vendte ud mod vinduerne og debet var det til den anden side - eller var det omvendt?? Han fortalte *det* helt indlysende at når man fjerner noget ét sted, skal det ind et andet, således at regnskabet hele tiden balancerer. Regnskab og bogholderi er åbenbart ikke helt så overnaturligt, som jeg havde troet, men mit kendskab til bogholderi stammede nu også udelukkende fra min tid i forsvarrets intendanturkorps. Jeg tænkte meget over det, jeg havde lært og kom så til erkendelse af at det i hvert fald ikke var *det*, vi havde programmeret.

Det var ikke populært, da jeg henvendte mig til min ledelse og anbefalede at vi reprogrammerede hele projektet. Der blev tænkt meget og analyseret en hel del. Resultatet blev at vi gik i gang *helt* forfra - denne gang med meget større succes. Jeg husker stadig tydeligt at det var et af de projekter, jeg var meget glad for at slippe, så meget mere som det faldt sammen med et andet stort projekt, der blev udviklet på System /360.

Apropos Royal's tournedos, jeg husker en overarbejds-aften, hvor vi var et par stykker, der besøgte Royal, for at få lidt fast føde, noget der i øvrigt skete ret tit på grund af meget overarbejde. Der var en amerikaner, der bestilte en tournedos. Servitricen spurgte ham rutinemæssigt om han ville have noget at drikke til. "*Ja tak, en kop varm chokolade*", var svaret. For en dansker var det lidt ejendommeligt at se ham gumle på sit røde lækre oksekød og drikke chokolade til. Selv for os IBM'ere, der jo ellers ikke er forvænt med drikkevarer. Onde tunger hævdede at IBM er forkortelse for **I**kke **B**ajere **M**ere. Min forkortelsesliste fortæller i øvrigt at

IBM også kan betyde: "*International Brotherhood of Magicians*", og det passer måske rigtig godt med det vi gik og lavede.

Arbejdet var ikke mere helt som før. Man kunne nu få pladehukommelser til 1401 - 1311 tror jeg de hed. Det var en stor fordel, men det gjorde samtidig livet mere kompliceret. Pladehukommelser styrer ikke sig selv, så der var behov for systemrutiner til at hjælpe med dette.

Jeg husker en opgave, der havde behov for tilfældig disk adgang (random access). Den blev løst før vi fik diske. Man lavede så 4 ens magnetbånd med data. Programmet søgte data skiftevis på et af båndene medens de tre andre spoledede tilbage. Jeg ved ikke hvor effektivt det var, men det så imponerende ud. Computere laver en masse, men det er *meget* sjældent der rigtig er noget at se på, men denne opgave var *virkelig* en fryd for øjet, båndene spoledede frem og tilbage i en rasende fart.

Rutinerne til styring af pladehukommelserne var den første spæde begyndelse til styresystemer/operativsystemer. Og styresystemer syntes jeg lød spændende - *meget* mere spændende end almindelig programmering. På den tid var det nok muligt at dyrke sine interesser, hvor heldig kan man være???? Titlen systemprogrammør var dog endnu ikke opfundet, men pyt, jeg har aldrig været særlig interesseret i titler.

System /360

I midten af tresserne annoncerede IBM **System /360**. Denne maskine blev annonceret som en universal maskine, der både skulle kunne benyttes til processtyring, kunne køre kommercielle og videnskabelige opgaver. Kommunikation mellem computere var vist endnu ikke opfundet, men kommunikation kræver nok ikke særligt meget af computeren. Det var en hel ny arkitektur der har vist sig at være særdeles holdbar, det er en videreudvikling af denne arkitektur, der stadig anvendes i IBM's store værtsmaskiner.

System /360 maskinerne var trediegenerations computere. Man var nu begyndt at afløse transistorerne med integrerede kredse. Jeg husker stadig reklamen med et fingerbøl, der var fyldt med 50.000 små transistorer.

System /360 maskinerne havde både binære og decimale instruktioner. Hukommelsespositionerne bestod af bytes á 8 bits, d.v.s. der var mulighed for 256 forskellige tegn i én position. To bytes udgjorde et halvord, fire bytes et ord og otte bytes et dobbelt-ord. De binære instruktioner arbejdede med halv- hel- og dobbeltord og da maskinerne af natur er binære, er binære instruktioner langt de hurtigste. Maskinerne arbejdede med variabel instruktionslængde og variabel feltlængde, det var kort sagt *meget* alsidige maskiner.

Noget af det revolutionerende ved /360 var dets interruptsystem, der gjorde det muligt at køre flere programmer samtidig. Interrupt er afbrydelse af instruktionssekvensen. Et af de vigtigste registre i maskinen var PSW (**P**rogram **S**tatus **W**ord). PSW indeholdt adressen på den instruktion man var i færd med at udføre, længden af instruktionen (så man kunne finde den næste) samt hvilke interrupts, der var tilladte, man kunne naturligvis ikke acceptere et interrupt af én type medens man var ved at behandle et andet af samme type.

På faste pladser i maskinens hukommelse, var der nye og gamle PSW's for de interrupts, der kunne forekomme. Når der kom et interrupt, gemte maskinen det aktuelle PSW (så at man kunne vende tilbage til den proces, man var i gang med), og erstattede det med det nye - svarende til behandlingen af det interrupt der var indtruffet, således at man direkte fortsatte i den kontrolprogram rutine, der gav den ønskede service. Interrupt forekom, når en input/output enhed var færdig, når man fra et program ville have service fra systemet, som f.eks. starte en input/output operation, når et tidsinterval var udløbet eller naturligvis, når der kom en maskinefejl. Når et interrupt var færdigbehandlet, var det systemets opgave at beslutte, hvilken rutine/program, der derefter skulle køre.

Det viste sig hurtigt at man med stor fordel kunne lade input/output-bundne programmer køre med høj prioritet, for de lange I/O tider gav megen ventetid, som så kunne udnyttes af mere proceskrævende programmer.

En af DCS's første opgaver på System /360 var opdatering af et tapekartotek, som bestod af ca. 25 magnetbånd. Jeg blev projectleder sammen med en systemkonsulent. Opgaven blev løst i COBOL - nu havde man fået flere high-level

sprog at vælge imellem og COBOL (**CO**mmun **B**usiness **O**riented **L**anguage) var skabt til kommerciel databehandling. Man påstod at det lignede almindeligt engelsk, men det er jeg nu ikke helt enig i, men det er dog rigtigt at det er *muligt* at skrive et COBOL program, så det minder (lidt) om engelsk.

System /360 var en ny teknologi og COBOL oversætterne var naturligvis også nye. Nu er der det ubehagelige ved nye ting at de ofte (altid?) er fejlbehæftede. Og fejlbehæftede *var* de første /360'ere og det *var* både styresystemerne og COBOL oversætterne også (altså fejlbehæftede), så det var et noget (meget?) ustabilt miljø at arbejde i, altså igen noget pioner-arbejde blot nu på et højere plan. For at bøde på disse store problemer, var systemerne udstyret med en facilitet man kaldte CHECKPOINT/RESTART. Idéen var at man med visse mellemrum tog et checkpoint (f.eks. når man kom til slutningen på en rulle magnetbånd). Checkpointet var et øjebliksbillede af systemet. Hvis noget gik galt, skulle man så kunne starte kørslen op fra det seneste checkpoint (RESTART) i stedet for at starte helt forfra. En yderst fornuftig (livsnødvendig??) tankegang i et ustabilt miljø.

Den før omtalte kørsel tog et døgn eller mere. Et døgn var *meget* lang tid for et program i de dage - måske for lang. Naturligvis indrettede vi vore programmer til at køre med checkpoint, og det virkede helt fint. "CHECKPOINT xx has been taken", var en meddelelse, vi så mange gange, meeeen der var den hage ved det at restart *ikke* virkede. Så vidt jeg husker var problemet at man ikke havde standset al aktivitet og tøvet et øjeblik, før man lavede checkpoint, det betød at der måske var ventende interrupts, der ikke blev behandlet, og et ventende interrupt kan man ikke gemme i et checkpoint.

Det var *meget* sørgeligt at komme ned i maskinstuen og se operatørerne stå og banke deres hoveder ind i væggen fordi programmet var gået ned, netop som det skulle til at udskrive slut-totalerne fra kørslen, det betød nemlig at man skulle køre om - alle disse 24 timer eller mere. Dels var det træls at skulle skifte alle disse magnetbånd endnu en gang, men det var også uheldigt fordi der ikke var for megen maskintid, så der var altid andre, der stod og trippede for at komme til.

Jeg var ikke selv involveret i programmeringen - heldigvis - jeg fungerede som en slags projektleder sammen med en 'rigtig' systemkonsulent. Det var spændende, udfordrende (vi skulle jo lære det hele) og til tider *meget* hårdt. Maskintid var jo også et problem, alle skulle jo arbejde med disse nye maskiner for at lære dem at kende, og da Danmark er et lille land fik IBM i Danmark ikke en større maskinpark fra starten. IBM Danmarks første /360 havde, så vidt jeg husker, 16 KB hukommelse. Dette er skrevet på en PC med 32MB hukommelse den er altså ca. 2000 gange større!!!

På trods af at jeg ikke selv skulle programmere, blev jeg sendt på et COBOL kursus. Kurset varede en uge, men jeg var ude at rejse (jeg husker ikke hvorhen eller hvorfor), så jeg gik glip af de første 2-3 dage. Det medførte naturligvis at jeg ikke begreb meget af, hvad man snakkede om på resten af kurset. Det tog jeg roligt,

det var jo ikke meningen jeg skulle kode, men da jeg til den afsluttende test scorede 100 points af 100 mulige (vistnok som den eneste på kurset), fik jeg fat i instruktøren og fortalte ham at hans test ikke var meget værd. Det var han ikke enig i, men jeg aner stadig ikke, hvordan man skriver COBOL programmer, og jeg tror jeg kunne have klaret testen selv om jeg *slet* ikke havde været på kurset, så jeg forstår ikke rigtig hvad det var man testede. Vi måtte benytte manualen, og alle spørgsmålene kunne findes i indholdsfortegnelsen, så det var egentlig meget let. Selv nu, længe efter jeg er blevet pensioneret, tror jeg stadig jeg ville kunne klare denne test uden fejl.

Jeg husker specielt én morgen. Jeg vidste der var travlhed på maskinen og problemer med programmet, så jeg sagde til min kone at det måske kunne trække ud, før hun så mig igen. Da jeg kom til IBM, fik jeg at vide at man havde skaffet os testtid i Stockholm!! Vi havde endnu ikke telefon hjemme (dengang kostede telefoner mange penge, og vi var nygifte), så jeg bad en sekretær om at sende et telegram til min kone og orientere hende. Hun skrev i telegrammet: 'Preben er taget til Stockholm på ubestemt tid. Han kontakter dig i morgen'. Hun fortalte ikke mig, hvad hun havde skrevet eller hvordan jeg skulle kontakte Kirsten, så da jeg ikke reagerede, udsendte Kirsten en efterlysning - dog heldigvis kun i IBM. Da jeg kom hjem fik jeg omgående besked på at ryste op med en telefon, hvad jeg selvfølgelig må erkende er en rimelig (uundværlig) ting for en EDB-mand - Men den har godt nok kostet familien formuer siden

Inden vi fik telefon havde vi flere gange prøvet at blive vækket på ukristelige tidspunkter af taxachauffører med telegrammer - altid med opfordring om at ringe til IBM. Det er ikke specielt morsomt at forlade sin seng midt om natten, klæde sig på og bevæge sig ud i kulden (det er vist altid koldt om natten, når man bliver jaget ud af den lune seng) for at finde en telefonautomat, der dog heldigvis var placeret tæt på vor daværende bolig.

En tur til Stockholm lyder måske nok spændende, det var første gang IBM sendte mig ud af landet, men vi fløj ved titiden om aftenen, arbejdede hele natten, sov næste dag, arbejdede endnu en nat og så hjem igen tidligt om morgenen. Det var ikke meget vi fik set af Stockholm ved den lejlighed. Jeg var ikke sætlig glad for Stockholm i begyndelsen, nok fordi jeg nåede at komme derop flere gange uden at få lejlighed til at se noget som helst af byen. Det er *ikke* rart. Man bliver ofte spurgt: 'Har du været i Stockholm, synes du ikke der er smukt???' og må så svare: 'Det aner jeg ikke, jeg har ikke set meget andet end IBM maskinstuer og hotelværelser'.

Der var en anden dag, jeg har lidt svært ved at glemme. Systemkonsulenten og jeg startede som så ofte før med en kop kaffe og et stykke smørkage i en nærliggende kafeteria. Formålet var at udveksle oplysninger om, hvad der var sket med vort projekt siden vist sidst havde set hinanden. Vi blev enige om at mødes med kunden, så vi satte os i hans FIAT 600 og drog til Hillerød. Sammen med kunden fik vi

defineret et problem i et program. Programmøren, der boede i Roskilde, var syg, så vi kørte til Roskilde og smed programlisterne på hendes dyne og bad hende finde en løsning. Derefter gik vi hjem til mig for at få frokost, vi boede bekvemt nok i samme bebyggelse. Efter frokosten hentede vi rettelserne og ringede til IBM for at bestille noget testtid. Da vi kom til IBM sidst på eftermiddagen, fik vi udleveret et par flyvebilletter til Sønderborg og en mand til at hjælpe os at slæbe, vi skulle nemlig til en stor virksomhed på Als for at teste om natten. Flyveturen blev en af de smukkeste, jeg har oplevet. Hen over det Fynske øhav ved solnedgang. Vi kom tidligt til Als, så der blev tid til en tournedos på Dyvig kro - en *meget* smuk sommeraften, hvor vi i det fjerne kunne høre fiskekutternes sagte dunk dunk dunk. Vi begyndte at teste ved 10-tiden. Det gav lidt af et problem, da maskinen havde en anden kanal-adresse end vort styresystem var vandt til. Heldigvis var der en af de *rigtige* teknikere, der hurtigt kunne bygge maskinen om - og lære os at fjerne ombygningen, når vi var færdige, maskinen skulle jo gerne fungere normalt næste morgen af hensyn til ejeren. Alt gik fint til klokken fire om morgenen. På det tidspunkt blev det tydeligt at der stadig var store problemer. Morgenflyveren til København gik ved 6-tiden - det var igen en utrolig smuk flyvetur. Så skulle vi lige et smut til Hillerød og aflægge rapport til kunden. Derefter fik vi en højst tiltrængt søvn. Det var en meeeget lang dag specielt når man tager det ringe udbytte i betragtning.

Testtid var i øvrigt et konstant problem, vi testede på næsten alle tidspunkter af døgnet og vi kom adskillige gange til Stockholm, et par gange var det i Malmø, vi besøgte en del kundeinstallationer og der var en enkelt tur til Hamborg. Hamborg og Malmø mener jeg nu ikke *jeg* var med til.

En tidlig morgen - ved fire -tiden, ringede telefonen. Det var min SE kollega - den anden projektleder, der var i Malmø for at teste. Han var kommet i tanke om at det kunne være morsomt at få konstateret om det var muligt at ringe fra Malmø til Roskilde med selvvalg. Det var det altså!!! Heldigt for ham at man ikke kan være voldelig igennem en telefon.

Der var en anden kørsel som jeg tydeligt husker, selv om jeg ikke selv havde noget med den at gøre. Det var et firma, der fik kørt en ugekørsel, der ofte varede fra fredag eftermiddag til onsdag morgen!! Resultatet af denne kørsel var *store* mængder af papir, så store at det næsten krævede en lastvogn til transporten. Jeg har ofte spekuleret på, hvor meget af alt dette papir, der nogen sinde blev læst.

Det var nok en helt almindelig fejl i de dage. EDB (sælgeren) gav mulighed for at få en mængde spændende oplysninger, så det var fristende for kunden at sige: 'Ja tak' til det hele, uanset om det var noget han havde brug for. Jeg er sikker på at mange vigtige data er blevet overset, fordi de har været gemt blandt en umådelig mængde af ligegyldige oplysninger. Også her kan man med rette tale om begrænsningens kunst.

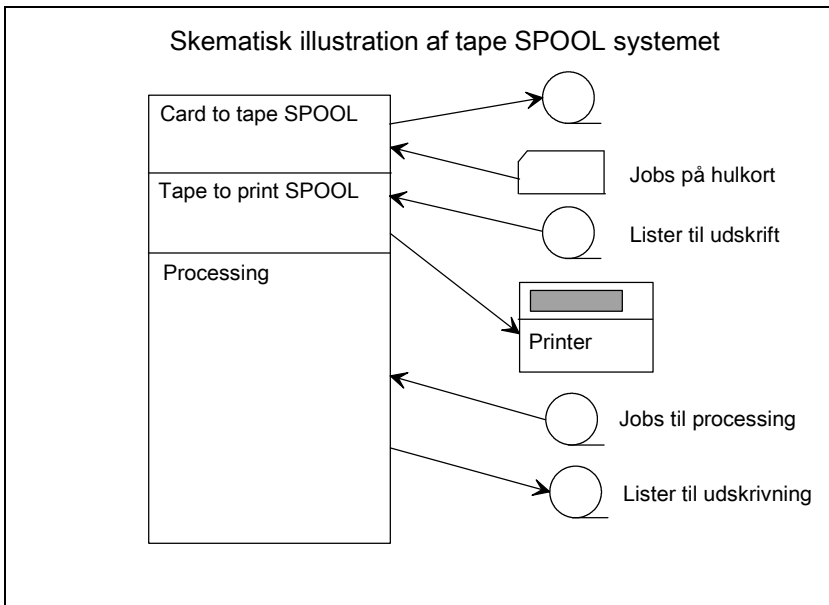
System /360 var langt mere kompliceret end de tidligere maskiner. De mange muligheder krævede naturligvis styreprogrammer (også kaldet operativsystemer), hvilket var noget *næsten* nyt. Til /360 havde man en række forskellige styre eller operativ systemer:

- **BOS - Basic Operating System.** Det blev vist aldrig rigtig populært. BOS var systemet, der var beregnet til de små - meget små - maskiner.
- **TOS - Tape Operating System.** Det var det første vi brugte i DCS, fordi det kom før DOS
- **DOS - Disk Operating System.** Det blev et system, jeg kom til at arbejde rigtig meget med. DOS var systemet, der var beregnet til mindre maskiner som den DCS havde. DOS lever stadig under navnet VSE (Virtual Storage Extended).
- **OS/PCP - Operating System Primary Control Program.** Dette var entry systemet til de større/store maskiner. OS/PCP var det eneste OS-system, der var tilgængelig fra starten.
- **OS/MFT - Operating System/Multiprogramming with a Fixed number of Tasks**
- **OS/MVT - Operating System/Multiprogramming with a Variable number of Tasks.**

I begyndelsen havde vi kun BOS, TOS og OS/PCP. BOS og OS/PCP kom jeg aldrig til at arbejde med, så i begyndelsen var det TOS, der blev brugt af DCS. TOS og DOS var principielt det samme system, forskellen var at TOS systemet var placeret på tape, medens DOS var placeret på en disk. Naturligvis er en disk langt mere hensigtsmæssig end tape til et operativsystem, så TOS var kun en midlertidig løsning, men diske var endnu ikke så almindelige den gang, og så var de dyre. Som tidligere nævnt var jeg mere interesseret i styresystemer end i programmering, så jeg fik fornøjelsen af at installere og arbejde med TOS og senere med DOS.

Nogle af de store landes DCS, havde i fællesskab udviklet et SPOOL system til TOS. Idéen med et SPOOL system var at der skulle køre et par ekstra programmer. Et input program, der skulle læse jobs fra kortlæseren til et magnetbånd. Programmerne skulle så læses og læse deres input data fra denne tape i stedet for fra hulkort, resultaterne - altså det, man normalt ville skrive på en printer, blev i stedet skrevet på en output-tape, som så af et output program blev skrevet på printeren. Det var en vældig god og effektiv måde at køre på.

Input/output programmer kørte med høj prioritet - fordi de havde mange og lange pauser - disse pauser blev så udnyttet til at køre de egentlige opgaver, der til gengæld kørte hurtigere, fordi input/output ville foregå på de hurtigere medier. Resultatet var at både maskinens processor og input/output medierne blev udnyttet meget bedre, med andre ord vi fik en bedre effektivitet. Dette er begyndelsen på multiprogrammering, altså det at køre flere programmer samtidig - næsten.



Da jeg nu var blevet den, der var interesseret i systemprogrammerne, fik jeg fornøjelsen af en tur til Milano - min første udlandsrejse for IBM og min eneste tur til Milano (bortset fra en mellemlanding langt senere). Det var i april, og Danmark havde været dækket af et tykt lag sne til at par dage før vi skulle af sted, Milano havde derimod herligt forår med kastanier med lys på.

I Milano skulle vi, fra de små lande, præsenteres for dette smarte DCS SPOOL system, så vi kunne bringe det hjem, så vi alle kunne få fordel af det. Det blev præsenteret af en italiener. Jeg husker ham tydeligt, han havde tilbragt en hel del tid i Poughkeepsie (det ligger lidt nord for New York - og jeg tror nok man skal have været der, for at kunne stave til det). Han havde et vældigt ordforråd, men en meget dårlig udtale (sydeuropæere er ikke så gode til engelsk som os fra nord), og da han talte med ægte italiensk temperament - altså lynende hurtig - var han slet ikke let at forstå.

En af de ting jeg lærte i Milano var at en 'Fernet Branca' er vældig god for en 'feriemave'. Det var bartenderen på hotellet, der syntes jeg så noget brugt ud - sådan følte jeg mig også. Det er den eneste gang, jeg har prøvet denne 'medicin', men den virkede altså hurtigt og fint.

TOS's SPOOL system var spændende, sådan noget havde jeg ikke haft med at gøre før, og SPOOL systemer blev snart en meget vigtig del af min tilværelse, men meget mere herom senere.

Jeg husker stadig jeg var lidt bekymret ved hjemrejsen, at udskifte Milano's dejlige forår med det uvenlige danske aprilsvejr synets jeg ikke lød rart. Heldigvis er det danske vejr aldrig helt som man forventer, så jeg landede i Kastrup i det dejligste danske (næsten) sommervejr, så overgangen var altså slet ikke så slem, som jeg frygtede.

Omsider blev TOS erstattet af DOS, det var virkelig en forbedring. Nu lå styresystemet på en disk, i stedet for en tape. Den indlysende fordel var naturligvis at det var meget hurtigere at få fat i det man skulle bruge. Det er jo langt hurtigere at flytte en disk-access-arm end det er at spole en tape, desuden slap man for en hel masse tapehåndtering. En maskine havde typisk 4-6 diske (der kunne udskiftes). Sådan en disk indeholdt ca. 7.5 millioner karakterer, så hvis man havde 6 af dem, havde man masser af plads, det *havde* man virkelig. DOS'en blev hurtig *min* baby. Jeg var nu hastigt på vej ind i jobbet som system programmør - skønt titlen endnu ikke var opfundet. Jeg husker der kom en hurtig fyr fra det store udland. Han fortalte os hvordan man skulle generere sådan et system (d.v.s. tildanne systemet til specifikke formål og maskiner) og vistnok også lidt om vedligeholdelsen. Jeg synes det var vældig spændende. Styresystemet skulle naturligvis arbejde tæt sammen med maskinen, så arbejdet med styresystemet måtte nødvendigvis medføre at man lærte en hel del om maskinens virkemåde, og teknik interesserede mig også. DCS's SPOOL system kom naturligvis hurtigt i en DOS udgave, og da dette system var blevet meget vigtigt for DCS, drog jeg ud i verden for at få det nye system præsenteret. En af ulemperne ved sådan noget 'privat' system kode er, at det skal opdateres hver gang standard systemet bliver opdateret, så vi kunne ikke skifte vor DOS til nyeste udgave, før SPOOL systemet var blevet tilpasset til det nye system. Dette var naturligvis også en fordel, for det betød at vi hver gang måtte rejse et eller andet spændende sted hen for at lære hvordan systemet *nu* virkede. Samtidig fik vi også lært nogle udenlandske IBM'ere at kende, og vi lærte også noget om, hvilke erfaringer man havde andre steder og at deres problemer stort set var de samme som vore, men erfaringsudvekslinger er nu en gang meget nyttige. En anden fordel var naturligvis de skattefri varer man kunne bringe hjem som *souvenirs*.

Da diskene kunne udskiftes - det var en pakke man satte ned i en boks - var det naturligt at have en ekstra DOS, hvis der nu skulle ske noget. Jeg tror ikke vi havde hørt om Murphy dengang, man hans love gjaldt alligevel, det er jo typisk for 'naturlove'.

Linieskriverne (printerne) var temmelig smarte. Der var et låg over dem, formodentlig for at beskytte dem eller operatøren mod støv og for at dæmpe støjen. Når der ikke var mere papir i printerne, åbnede låget sig automatisk, så det var lettere og hurtigere for operatøren at sætte nyt i.

En dag da man kørte BOS på maskinen, lå vor DOS ovenpå printerens og kopien lå bagved. Så slap papiret uheldigvis op og låget åbnede sig, hvorefter den ene DOS

faldt ned og smadrede sig selv og den anden. Joh, Murphy's love gjaldt bestemt også den gang. Vi brugte en laang nat til at lave et nyt system. Hen på morgenen, da vi var færdige, kom en systemkonsulent og hev en bunke hulkort op af lommen med en elegant bevægelse. Det var et backup/restore program. Det var første gang vi hørte om sådan noget, men det var lige hvad vi havde brug for. Med det program kunne man kopiere indholdet af en disk op på en tape, eller man kunne lægge en tapes indhold ned på en disk. Nu var det et dumt tidspunkt på dagen (morgener er næsten altid dumme, når man har arbejdet hele natten), så et eller andet gik helt galt. Det var vistnok noget med adresseringen af disken og tapeen. Resultatet var at vi lagde et arbejdsbånd's indhold ned på den DOS pakke, vi havde brugt en hel nat til at lave. Slet ingen tvivl, Murphy's love *var* gyldige - også i de dage. Det var bedrøveligt at se den stakkels SE'er, sagte gå bort i solopgangen med et trist udtryk i ansigtet..

Efterhånden blev vi dygtigere, vi fik etableret et fornuftigt produktionsmiljø, hvor der blev løst mange forskellige opgaver. Da vi var et datacenter, der kørte opgaver for mange forskellige kunder, havde vi mange flere programmer end systemet var beregnet til. Sådan en DOS og dens programbiblioteker lå på én diskpakke, så der var begrænset plads. Nu havde vi heldigvis adgang til en pulje af IBM udviklede programmer - der oven i købet var gratis, også for kunder. Et af disse programmer gjorde det muligt at have private programbiblioteker, således at man kunne have programmer på så mange diske, som man ønskede - noget der er helt naturligt i dag. Det virkede fint, man skulle bare kalde et specielt program, der så læste det program, man ville eksekvere ind i maskinen. Der var endog mulighed for at lave programmer med overlay altså det at skifte inaktive dele af programmet ud med aktive - noget man ofte havde brug for før vi fik virtuel hukommelse.

Vi var i mellemtiden blevet to mand til at vedligeholde DOS - det var blevet et *meget* vigtigt system for os. En sommer, hvor systemerne opførte sig pænt (det betyder at vi systemprogrammører havde god tid), besluttede vi at vi ville indbygge det private bibliotekssystem i DOS'ens supervisor. Vi havde hørt at man kun kunne skrive, teste og dokumentere tre linier kode om dagen, når man skrev systemprogrammer. Det mente vi var *alt* for lidt, vi var jo *dygtige*!! Vi gik i gang med krum hals. Det var ikke så simpelt som vi først troede, det er sådan noget aldrig, men vi begyndte at læse i system koden, for at finde ud af, hvad der foregik. Man kan ikke ændre noget ordentligt, uden at kende det grundigt. Vi lærte os altså DOS'en funktioner til bunds (næsten), og begyndte så at skrive og derefter at teste. Det blev et *meget* stort arbejde, faktisk tog det os det meste af sommeren. Men det lykkedes, og sjovt var det, og vi var bestemt stolte over vores arbejde. Det var en virkelig god og funktionel løsning, vi havde fået lavet. Og fordelene ved den var at altting nu virkede fuldstændig, som det var beskrevet i bøgerne - altså ikke mere noget med et specielt indlæsningsprogram og specielle overlay rutiner. Altting var

standard - altså set fra programmernes side. DCS i Danmark benyttede vort eget 'Private Library System' med succes i lange tider.

Omsider kom der standard support for private biblioteker, det gjorde naturligvis vore modifikationer overflødige, og man skal bestemt ikke køre med modificerede systemer, hvis man kan undgå det, fordi det er et meget stort arbejde at vedligeholde. Men det var med en vis tilfredshed og stolthed vi kunne konstatere at vore modifikationer faktisk var en hel del bedre og mere elegante end den support, der blev indført som standard i DOS. Desværre kan jeg ikke mere huske detaljer, men vi var en smule skuffede, da vi så hvad 'de' var kommet frem til, allerede på det tidspunkt gik det op for mig at vi i lille Danmark var mindst lige så dygtige som mange af eksperterne fra det store udland.

Da vi var færdige, og så tilbage, måtte vi erkende at vi nok ikke havde produceret mere end de ca. tre linier kode om dagen som man hævdede var standarden, men det var i hvertfald *kvalitetskode!!!* Der var nok ikke megen dokumentation, men det blev da beskrevet, og vi havde haft en særdeles morsom og lærerig sommer.

I DataCenteret var data sikkerhed naturligvis yderst vigtigt, så vi tog regelmæssigt back-up af alle vore udskiftelige disk-pakker. For at være *helt* sikker, prøvede vi altid at kontrol læse tapene, det ville jo være uheldigt, hvis der var en fejl på én af dem, hvis der blev brug for den.

En dag skete det så. En af diskene gik i stykker. Ikke noget problem, operatørerne fandt den nyeste back-up tape og . . . Den kunne *ikke* læses. Så gik teknikerne til bekendelse. De havde for nylig haft alle tape-stationer inde på deres værksted for at give dem et rutinemæssigt eftersyn, herunder en justering. Uheldigvis havde båndstationen, der blev brugt til back-up kørslen været ude af justering, og derfor kunne tapen ikke mere læses. Teknikeren måtte altså - meget mod sin vilje - justere tapen tilbage, bare medens vi læste back-up tapen. Datasikkerheden var altså god nok fordi vi på trods af problemerne var i stand til at komme videre.

Selv om /360 var en ny teknologi, eksisterede der stadig nogle 'gamle' 1401 programmer. Heldigvis kunne /360 simulere 1401. Det var noget med at man kunne definere instruktioner ved hjælp af en slags mikro-programmering - og 1401 instruktioner kunne altså efterlignes, men da det nu kørte på en mere moderne maskine, kørte det naturligvis meget hurtigere end på 1401 - altså i teorien.

En dag talte jeg med en programmør, der havde ansvaret for en 1401 FORTRAN kørsel på vor /360 maskine. Han fortalte at det kørte fint, men desværre tog det meget længere tid end på en 'rigtig' 1401. Sådan noget syntes jeg lød forkert (læs rablende tåbeligt), så jeg gav mig til at bore lidt i sagen. Jeg fik fortalt at der kom en fejlmeddelelse et par gange i begyndelsen af kørslen, men derefter kørte alt fint. Jeg så efter i bogen med fejlmeddelelser, og jeg fandt ud af at meddelelsen betød at der blev lavet beregninger på tal af forskellige formater - så en konvertering var nødvendigt. Yderligere stod der at den pågældende meddelelse kun ville blive vist

de første 10 gange. Da vi rettede denne lille fejl, kom programmet - som forventet - til at køre *meget* hurtigere end det gjorde på 1401. Den pågældende konvertering af tal blev gjort utallige gange og tog derfor rigtig lang tid. Moralen må vel være at man altid skal slå en meddelelse op i manualen, hvis man ikke er *helt* sikker på, hvad den betyder - og helt sikker er man vel aldrig.

I begyndelsen var /360 maskinerne ikke særlig stabile. Jeg husker der skulle holdes en *meget* vigtig demonstration - en af den slags der *ikke* måtte gå galt. Det var noget med at fremvise en skærmterminal - den gang noget *helt* nyt. En kreativ fyr fandt på en backup-løsning. Vi kørte en prøve på demonstrationen, og opsamlede signalerne til skærmen på en almindelig kassettebåndoptager. Det viste sig at være en rigtig god idé. Maskinen besluttede sig for *ikke* at køre da den absolut skulle. Sælgeren holdt sin demonstration, men det var båndoptageren, der bestemte, hvad der skulle stå på skærmen - maskinen var *slet* ikke impliceret. Kunden mærkede heldigvis ikke noget, men det må have været lidt af et pres for sælgeren, der ikke blot skulle sige de rigtige ting, men også skulle afpasse tempoet efter båndoptageren, der jo bare kørte.

Hen af vejen blev de store lande større - målt med IBM målestok - hvilket betød større og kraftigere maskiner. Og større maskiner betød behov for større og mere avancerede styresystemer, så de store lande skiftede efterhånden fra DOS til en OS variant. Man kunne på det tidspunkt vælge mellem OS/MFT eller OS/MVT, hvor MVT var langt det mest avancerede.

MFT var 'Multiprogramming with a Fixed number of Task', hvilket betød at systemet skulle defineres med oplysninger om hvor mange programmer man ville køre samtidig, og med hvor megen plads, der skulle være til hvert program. I de dage var man meget afhængig af systemets hukommelse.

MVT var 'Multiprogramming with a Variabel number of Tasks'. Systemet kunne altså selv starte nye tasks, hvis der var ressourcer til det. Det var altså et meget mere fleksibelt system.

Dette skift betød at de store lande ikke mere var interesserede i det DCS udviklede SPOOL system. Disse landes sidste bidrag til dette system var en workshop, der blev holdt i en forstad til Stuttgart, hvor systemet blev overdraget til os 'små'. Vi fik fortalt, hvad og hvordan og fik koden udleveret. Derefter måtte vi selv modificere det, så det passede til kommende udgaver af DOS. Jeg tror dog kun vi nåede at modificere DCS SPOOL til én ny DOS release, så fik vi selv behov for at skifte til de større systemer.

Jeg ved ikke om det var her det begyndte, men efterhånden blev der et vældigt godt samarbejde mellem de interne IBM systemprogrammører på tværs af landegrænser, et samarbejde jeg tror var til stor nytte både for IBM og for 'os på gulvet'. Vore rejse-aktiviteter var ikke store, så det var begrænset hvor tit vi så hin anden, men dengang kunne vi skrive rigtige breve - altså dem på hvidt papir ligesom vi kunne

tale i telefon, det var i øvrigt den eneste mulighed vi dengang havde for at kunne kommunikere..

DCS i København blev efterhånden ret stor, og vore maskiner voksede i takt hermed. Hver gang der kom en ny enhed, skulle den genereres ind i DOS'ens supervisor. En dag jeg lavede en triviell ændring, gik det *helt* galt. Systemet kunne overhoved ikke køre. Mærkelig, for det var bestemt ikke noget stort jeg havde lavet. Efter at have gransket det sædvanlige storage dump, gik det op for mig at problemet var at supervisoren var blevet *for* stor. Da man byggede DOS, var der ingen der kunne drømme om at en supervisor kunne blive mere end 32K (det var for øvrigt størrelsen på vor første /360 maskine). Derfor havde man lagt adresser ind i halvord - altså 2 bytes. I et halvord kan man kun have adresser op til 32K altså 2¹⁵, der er 32768. Dette minder lidt om Bill Gates: *'Der er aldrig nogen PC, der får brug for mere end 640K hukommelse'*.

Det var nok den gang vi indså at vi måske var ved at vokse ud af DOS som styresystem. DOS havde tjent os trofast i mange år, men nu var den altså ved at være for lille.

Derfor skiftede vi, først til OS/MFT, men senere - så vidt jeg husker ret snart - til OS/MVT, der var det mest avancerede system, man kunne få på det tidspunkt.

OS, alle systemerne, arbejdede på den måde at et readerprogram læste jobs, og placerede dem i en job-kø. Derefter blev de behandlet, en initiator læste jobbene fra køen, fandt de nødvendige ressourcer og startede de tilhørende programmer og udførte dem. Under eksekveringen blev eventuelle lister placeret i en print kø, hvorefter et writer-program læste dem fra printkøen og skrev dem ud på en linieskriver. Dette var et helt traditionelt SPOOL system, ganske som tidligere beskrevet. Hvis man kørte OS/PCP, kørte kun ét program ad gangen - altså ingen multiprogrammering. Hvis man kørte MFT eller MVT, kunne reader, writer programmer køre samtidig med applikationsprogrammer, hvilket naturligvis gav en langt bedre udnyttelse af maskinerne. Hvis man havde behov for remote job entry, altså behov for at sende jobs fra arbejdsstationer, der var koblet til systemet via telelinier, ville en OS-reader kræve mere end 100K. Det var *meget* hukommelse i de dage, så vi begyndte at interessere os for alternative muligheder. Vi fandt ét: HASP. HASP blev min næste store computerkærlighed.

HASP

Da vi var to systemprogrammører til at passe maskinerne, og stod for at skifte operativsystem, var det naturligt at vi forsøgte at dele arbejdet i mellem os. Da jeg nu havde arbejdet rigtig meget med SPOOL-systemer, var det oplagt at jeg fik ansvaret for SPOOL-systemet i vort nye operativsystem, og dette SPOOL system blev altså HASP. HASP er en forkortelse af **H**uston **A**utomatic **S**pooling and **P**riority System. Det var, som navnet antyder, udviklet i Huston, Texas. I Huston havde man (det *er* der vist for øvrigt stadig) et Space Research Center, og i sådan et foretagende har man brug for meget kraftige computere.

Man anskaffede sig altså et par af de største System /360, men uheldigvis var der på det tidspunkt kun mulighed for at benytte operativsystemet OS/PCP (ikke bare PCP, men en meget tidlig udgave af PCP). Og det var et system uden mulighed for at køre flere programmer samtidig, og det var næppe en effektiv udnyttelse af så store maskiner, så en lille kreativ gruppe med en 'Tom Simpson' i spidsen, skrev det geniale SPOOL-system, der kom til at hedde HASP.

HASP overtog kontrollen over maskinen ved hjælp af den teknik der kaldtes PSW stealing. Man erstattede NEW PSW for de interessante interrupts således at HASP fik kontrollen i stedet for styresystemet. HASP sorterede så disse interrupts og behandlede de af dem, der var interessante for den selv, resten blev overgivet til selve operativsystemet ved at loadet det oprindelige NEW PSW. Denne teknik er ret snedig, fordi den kan benyttes af flere programmer samtidig. Naturligvis er det, det sidst startede program, der først får kontrollen. HASP havde sin egen rutine, der kunne afgøre hvilket program/rutine der skulle køre, når HASP fik kontrollen. Det kunne være HASP reader(e) eller writer(e) eller hvis HASP havde gjort, hvad den skulle, blev kontrollen overgivet til selve styresystemet (eller måske et applikationsprogram). På den måde kunne man køre en slags multiprogrammering også under OS/PCP. En genial løsning, som de naturligvis var meget glade for i Huston. Heldigvis stillede IBM HASP til rådighed for alle kunderne, så mange flere kunne udnytte dets store fordele.

HASP's hovedopgave var at håndtere langsomme enheder, som korthuller(e), kortlæser(e) og printer(e), så mange af disse som man havde til rådighed. Disse enheder kunne både være lokale eller fjernt placerede koblet til maskinen med telefonlinier og modems. Remote Job Entry blev ret meget anvendt også i DCS, der den gang havde afdelinger i Odense, Aalborg og Århus. For at udnytte telefonlinierne så effektivt som mulig, havde HASP udviklet sin egen specielle linieprotokol, kaldet HTAM (**H**ASP **T**elecommunication **A**ccess **M**ethod). Jeg er ikke kommunikationsspecialist - slet ikke, men jeg har fået forklaret at det smarte ved HTAM var at når der skulle sendes kontrolsignaler ud på linien f.eks. når kommunikationen skulle ændre retning, sendte man en pakke data med. På den måde opnåede man at kunne sende mange flere data meget hurtigere, da man jo sendte data i begge retninger næsten samtidig.

Der fandtes to 'add on' SPOOL systemer, der var tilgængelige for OS. ASP (Attached Support Processor) og HASP.

ASP var skrevet til at køre på to eller flere maskiner. Den ene tog sig af SPOOL og planlægning og den anden udførte jobbene. Jeg kan stadig undre mig over hvordan man skelnede mellem disse systemer i Frankrig, hvor et 'H' som bekendt ikke udtales. ASP var mere avanceret og større end HASP, men da vi kun havde én maskine, der ikke var voldsomt stor, var HASP nok det rigtige valg for os. Yderligere var HASP vistnok meget mere elegant - jeg ved meget lidt om ASP, så jeg kan ikke rigtig bedømme det. Jeg mener DataCentralen var den eneste danske installation, der benyttede ASP.

HASP indeholdt, hvad der svarede til OS's reader og writer, både lokal og remote. Et typisk HASP system fyldte mindre end 50K i hukommelsen - altså langt mindre end OS's tilsvarende standard komponenter (en standard OS remote job entry reader fyldte langt over 100K). Da HASP samtidig krævede langt mindre procesorkraft end de tilsvarende OS komponenter er derfor forståeligt at HASP hurtigt blev særdeles populært. Og populært blev HASP sandelig. Jeg tror nok jeg stadig har 'The HASP Songbook', der f.eks. indeholdt sange som 'HASP Days are here again' og 'HASP belongs to everyone', the best things in life are free'. Disse sange var alle en hyldest til HASP og til HASP-guru'en Tom Simpson. Jeg tror ikke vi kendte begrebet guru dengang, det var i hvertfald ikke almindelig brugt, men naturligvis eksisterede der guru'er, det har der nok altid gjort.

Jeg begyndte naturligvis at sætte mig ind i HASP's muligheder, og de var mange, og mange af dem passede særdeles godt til DCS's behov. Nu var jeg omsider blevet rigtig systemprogrammør med titel og det hele. En af de store fordele ved *det* job er at man *skal* være specialist, ikke bare af navn, men sådan *rigtig*. Det bliver man ikke af sig selv, så man får eller må tage tiden, der er nødvendig, for at opnå et tilbundsående kendskab til de produkter man arbejder med. Der findes ingen uddannelser på *så* specielle områder, så en specialist på den slags områder må nødvendigvis være autodidakt.

Hvad er en specialist?? Joh, en specialist er en, der kommer til at vide mere og mere om mindre og mindre. Modsat en ekspert, det kommer til at vide mindre og mindre om mere og mere. Specialisten ved altså en hel masse om sit speciale og ikke ret meget andet, medens eksperten ved lidt om næsten alting. Dette er naturligvis min egen private definition, men jeg tror nok jeg har adopteret det fra en kollega, så vi er altså mindst to om den.

Det var en fornøjelse at sætte sig ind i HASP. HASP var nok det bedst skrevne program jeg har set og arbejdet med - måske indtil CMS's pipeline så dagen lys mange mange år senere. Det var vel-dokumenteret, logisk opbygget og havde et væld af gode funktioner. Det var dog ikke nok. HASP gruppen havde indset at det er svært at gøre alle glade, EDB folk er, eller de var dengang udpræget individualister, så systemet var indrettet så det var let at modificere/udbygge/tilpasse. Enkelte

steder i koden var der oven i købet kommentarer, der beskrev, hvordan man skulle bære sig ad med at ændre.

Hvis man f.eks. ønskede at indføre en ny kommando, var der en beskrivelse af hvilke kontrolblokke sådan en kommando skulle defineres i, og hvordan man kaldte den bagved liggende kode. Man fik beskrevet hvilke registre der gav de nødvendige oplysninger o.s.v. Hvis man fulgte disse retningslinier, ville ens modifikationer være i overensstemmelse med HASP's kodningsregler, og de ville umiddelbart kunne anvendes i, og overføres til evt. nyere udgaver af HASP.

DCS havde en del ønsker om faciliteter, der ikke var i HASP, så jeg gik i gang med kuglepenen. Nu var det ikke alting, der var så godt beskrevet, som indførelse af *nye kommandoer*, og da HASP var et stort og kompliceret system, var det nødvendigt at *træde* forsigtigt, før jeg gik i gang. Da jeg på denne måde kom godt rundt i koden, endte det med at jeg lærte HASP *rigtig* godt at kende. Det er nok det eneste område, hvor jeg uden at rødme kan hævde at jeg var den i landet, der vidste mest. Danmarksminister hva' ba' - ganske vist uofficielt, altså ingen udnævnelse - men alligevel. Jeg boltrede mig som en fisk i vandet. Det her var noget andet end almindelig programmering. Ganske vist skulle jeg være *mindst* lige så omhyggelig, fejl kunne jo få forfærdelige følger for hele systemet, men da jeg havde en fornemmelse af at det jeg gjorde var nyttigt, var det til at leve med. Hvis jeg skal være helt ærlig, må jeg erkende at der var nogle af mine modifikationer, der ikke var helt i overensstemmelse med god tone i HASP, men de var nødvendige, de virkede, og de opfyldte et behov for DCS.

Mine modifikationer fyldte efterhånden ca. 2000 hulkort, de var *naturligvis* ikke nummererede, så jeg var noget *knyttet*, da en piccolo tabte kassen med mine kort under transport med det resultat at de lå hulter til bulter. Han var en flink fyr og *meget* ked af det, så jeg sagde ikke noget - i hvertfald ikke ret meget!! Men jeg fik da lært at den tid, man bruger til at lave sekvens-numre kan være godt anvendt. Det tog mig adskillige timer at få kortene lagt i den rigtige rækkefølge, og sådanne timer kan nemt bruges til noget mere fornuftigt.

Jeg husker desværre ikke meget om de modifikationer, jeg lavede. Jeg lavede et par kommandoer, jeg husker ikke hvilke, og så lavede jeg en rutine til afregning af udskrivning. HASP havde rutiner til afregning af forbruget af forskellige ressourcer, men med hensyn til print, blev alt print afregnet - også selv om det ikke kom på papir, f.eks. fordi det blev sendt med telefon til en fjern station. DCS ville imidlertid gerne have en afregning af, hvad der kom ud på papir, så sådan én lavede jeg - det var imidlertid en af de modifikationer, hvor jeg ikke helt fulgte, hvad der var god tone i HASP. Jeg husker ikke, hvad der var galt, men jeg husker tydeligt at min samvittighed ikke havde det helt godt. Måske skal jeg ikke tage det så tungt, for modifikationen lavede *ingen* ulykker, og den opfyldte helt bestemt et relevant behov - og så var den vel også med til at styrke min anseelse, til gengæld gav den nok lidt mere besvær, når HASP skulle fornys eller have installeret rettelser.

Disse modifikationer gjorde at jeg kom meget rundt i systemet og jeg fik efterhånden et ganske godt ry i den forbindelse. Teknisk afdeling - altså dem der arbejdede med software - var normalt nogle man så op til, de var dygtige og havde adgang til de rigtige databaser, men når det gjaldt HASP, var det dem, der kom til mig for at få hjælp også indenfor deres område. Den var en rigtig dejlig fornemmelse.

Jeg husker et problem på NEUCC. Man kørte APL og HASP. APL (A Programming Language) var et system, der satte sig selv umådelig højt, altså APL frem for alt. Blandt andet for at køre hurtigt var måden APL opførte sig på ikke altid helt stueren. Når APL kørte, under MVS, lånte den til tider en tilfældig TCB (Task Control Blok). TCB var den kontrolblok, der styrede et task, normalt et program. Når MVS fandt en fejl i et task, blev det afsluttet ved hjælp af denne TCB. Hvis nu APL havde lånt HASP's TCB, og APL havde en fejl, ville MVS altså afslutte HASP. Men da nu nogle PSW's pegede på HASP-koden, og den altså ikke eksisterede mere, betød det at MVS havde slået sig selv ihjel - altså bogstavelig talt savet den gren over, den sad på. Så på trods af at problemet så ud som om HASP, af en eller anden mærkelig grund tilsyneladende sluttede unormalt, lå problemet altså i et helt andet program - nemlig APL. Jeg kendte HASP godt nok til at vide at den ikke fejlede på denne måde, men det var lidt tricky. Den slags problemer, sletter jo som regel sporene efter sig, og det er jo ikke let at vide, hvordan verden så ud lige før problemet opstod!!!! Heldigvis har operativsystemer normalt en tracetabel, der viser lidt om, hvad der skete - lige før.

Jeg husker stadig hvor stolt, jeg var, da jeg omsider havde fundet løsningen. Og så teknikerens beundrende og imponerede blik - joh, sådan noget oplever man ikke ret tit. Den pågældende tekniker blev senere programtekniker for VM, mit næste store arbejdsområde, så jeg kom til at arbejde rigtig meget sammen med ham, så vi fik alt i alt meget glæde af hin anden., noget der heldigvis tit er sket for mig med forskellige kolleger.

Jeg har et eksempel, der viser hvor stabilt et system HASP var. En systemprogrammør havde fået fat i et program, der skulle kunne rydde op på en disk. Programmet skulle kunne slette alle dataset, der ikke var systemdataset. Noget gik dog galt - som så ofte før. Det var nok en betingelse, der var blevet vendt 180°, med det resultat at netop alle systemdataset blev slettet. Da selve systemdisken, var der, hvor oprydning var mest påkrævet, var det naturligvis på den, programmet blev testet. Det her ligner Murphy igen: Alt hvad der kan gå galt, går galt. Programmet virkede meget fint og særdeles hurtigt. Alle systemdataset forsvandt øjeblikkelig. Det betød at der ikke mere var noget operativsystem i maskinen, meen HASP kørte videre som om intet var sket. HASP var ved at udskrive en fil, og det fortsatte den med indtil vi rørte ved konsollen, dette krævede nemlig en OS rutine, og da den ikke var der, døde maskinen omsider. Jeg er næsten sikker på det er den eneste gang jeg har set en maskine køre videre medens operativsystemet blev fjernet under den - og dog, men på *det* tidspunkt var det rigtigt.

Min succes med HASP blev ikke overset. Jeg tror det var i foråret 1973, jeg blev kaldt ind til min afdelingsleder og fik forklaret at man forsøgsvis ville prøve at sende nogle af medarbejderne fra DCS med på DP's konvention. Konventions var noget salgsafdelingen havde som belønning til de sælgere, der havde solgt hele deres kvota (derfor blev det ofte kaldt for 100% klubben) og til de DP systemkonsulenter, der havde gjort en ekstraordinær stor indsats.

Jeg var altså en af de heldige (dygtige??), og jeg fik at vide at vi skulle til Lissabon. Noget senere blev 'vi' kaldt til et møde hos afdelingschefen, der brugte megen tid på at forklare os at på sådan en konvention var der rigtig mange topchefer - altså dem fra det store udland - så vi skulle passe på at opføre os rimeligt pænt. Jeg husker tydeligt flyveturen til Lissabon, vi var vel nået over Hamborg, da samme afdelingschefs lettere slørede stemme lød i flyverens højtalere: *"Hvis man ska' tisse, ska' man ha' en drink med, for at komme forbi mig"!!!!* Det var altså den samme mand, der havde forklaret os, hvor vigtigt det var at opføre sig pænt!!

Sådan en konvention var en rimelig fornem sag. Vi ankom først på eftermiddagen og blev indlogeret på en fornemt hotel, to og to. Der var marmor badeværelse, balkon med udsigt til en lækker svømmepøl og i det fjerne Atlanterhavet omkranset af viftende palmer. Kort sagt lige til en turistbrochure.

Det var først i april, og vejret var som en behagelig dansk sommer. Der var en stor velkomstmiddag med masser af god mad og vine ad libitum - mit yndlingsmærke, så vi var vistnok mange, der næste dag havde lidt problemer i den store sal, hvor mødet foregik. På fortæppet var projiceret et psykedelisk billede, der bølgede - flimrede billeder og tømmermænd er virkelig ikke nogen god blanding. Musikken var The Beatles 'Here comes the Sun'. Det passede jo meget fint til det flotte vejr, der var udenfor.

Jeg fik set Lissabon og nogle af omgivelserne. Der var en portugisisk tyrefægtning specielt arrangeret for IBM (det var udenfor sæsonen). I en portugisisk tyrefægtning dræber man ikke tyren - ellers minder den meget om den spanske. Jeg fik nogle gode middage og en masse at drikke. Ja en aften nåede vi endog at komme ned i havnekvarteret på en temmelig tvivlsom knejpe. Vi var to, der ikke syntes det var så morsomt (selv om pigerne så meget søde ud), så vi endte med en flaske whisky på balkonen til mit værelse. Så blev verdenssituationen ordnet, som det sig hør og bør, og vejret var behageligt til skjorteærmer. Konvention gaven var et marmor askebæger. Sådan et er uforgængeligt, så jeg har det stadig, selv om der ikke er nogle rygere i familien.

Jeg var naturligvis ikke den eneste, der lavede HASP modifikationer. Brugergrupper udsendte regelmæssig lister over tilgængelige modifikationer. På en sådan liste fandt jeg en virkelig spændende modifikation, lige noget for DCS. Det var NIH (National Institute of Health) i Bethesda Pennsylvania, der havde udviklet et fælles SPOOL system til flere maskiner, der kørte HASP. Man kunne altså have

flere maskiner med HASP, der opererede med de samme SPOOL-kø'er. Da DCS nu havde fået flere HASP-maskiner, var det vældig interessant. Jeg prøvede energisk, det gjorde jeg virkelig, men i de dage var det slet ikke let at få IBM til at sende én til USA og da slet ikke for at lære om et kunde-udviklet program.

Udviklingen gik naturligvis videre. I begyndelsen af 70-erne kom der en serie nye maskiner og nye styresystemer. HASP blev en del af operativsystemet under navnet JES2. Jeg har stadig meddelelsen fra HASP gruppen, hvor de beklagede at de måtte lukke fordi HASP fremover ville blive til JES2. Meddelelsen har en bred sørgerand, og jeg tror at også de fleste kunder var bedrøvede. HASP var dog ikke død - slet ikke, men nu var det et officielt IBM produkt, og derfor meget mere stift og utilnærmeligt. Med andre ord, noget af morskaben forsvandt, nu var det ikke mere et produkt, man lavede modifikationer til. Det betød jo så for mig, at der ikke mere var behov for en HASP-specialist.

This page intentionally left blank.

VM/370

Der kom nu en af de helt store annonceringer, ikke bare en ansigtsløftning af de bestående maskiner, men en helt ny serie.

De nye maskiner hed System /370. Det var ikke en ny arkitektur, mere en videreudvikling af /360 arkitekturen, en af nyhederne var at maskinerne nu kunne benytte virtuel storage. Så vidt jeg husker var der dog ikke virtuel storage fra starten, men det var stadig det, det drejede sig om. Det er sket flere gange at nye annonceringer ikke fra starten har haft alle nyhederne - formodentlig fordi de ikke har været helt klar. Egentlig betryggende, det er ikke rart at få en masse smarte nye muligheder, for blot at konstatere at de ikke virker som de skal.

Sammen med de nye maskiner kom der en række nye operativsystemer:

- MFT blev erstattet en OS/VS1
- MVT blev erstattet af OS/VS2, der senere blev til MVS (**M**ultiple **V**irtual **S**torage)
- VM/370 - **V**irtual **M**achines noget helt nyt (næsten, det var en videreudvikling af CP-67, der imidlertid ikke var særlig udbredt)

Virtual storage var ikke noget nyt, det var ikke en IBM opfindelse, men ikke en gang i IBM var det nyt. Idéen med virtuel storage er den enkle (og fornuftige) at man kan bruge disk som supplement til maskinens hukommelse. Man opdeler hukommelsen i lag. Det øverste lag er hukommelsen i maskinen. Næste lag er hukommelse, der ligger på en disk, meget senere kom der endnu flere af disse lag. Man deler maskinens hukommelse op i sider (pages) á 4096 bytes. Når et program skal bruge hukommelse, bliver den tildelt i pages - så mange, der er brug for. Når der ikke er flere pages til rådighed, vil styresystemet flytte pages, der ikke har været brugt et stykke tid, ud på en disk, der er (næsten) altid dele af et program, der på et givet tidspunkt er inaktive. Styresystemet skal så naturligvis føre regnskab med, hvor en given hukommelsesdel for et givet program befinder sig på et givet tidspunkt, det sker ved hjælp af en masse tabeller. Man skal naturligvis også have en ny adresserings filosofi og man må skelne mellem reelle og virtuelle adresser. Maskininstruktionerne bruger naturligvis de reelle adresser, medens programmerne bruger virtuelle adresser. Resultatet af dette er at en maskine kan køre som om den har meget mere hukommelse end den rent faktisk har.

Hvorfor nu det??? Jo, processor-tid er det mest kostbare i maskinen, så det er vigtigt at processoren laver så meget som muligt. Det betyder der skal køre så mange programmer som muligt. Men programmer kræver hukommelse og hukommelsen er den næst-dyreste ressource, så det er smart at erstatte den med diskplads, da diskplads er billigere. Det *virker* faktisk. Selvfølgelig må et program fra tid til anden vente på at dele af dets hukommelse læses fra en disk til maskinens hukommelse, men det betyder ikke noget, hvis bare andre programmer kan køre imedens. Og skønheden i dette er, at programmerne ikke skal bekymre sig da hele styringen ligger i operativsystemet. Det er nu muligt at skrive programmer, hvor man ikke

skal bekymre sig om plads. Det betyder normalt hurtigere programmer, og i hvertfald hurtigere kodede programmer - altså billigere programmer, og det betød desuden at teknikken med overlay helt gik i glemmebogen.

Jeg har flittigt benyttet ordet virtuel. Det kommer naturligvis, som det meste af computer terminologien fra engelsk. Da jeg stødte på ordet, slog jeg det op i en ordbog, uden dog at finde en forklaring jeg kunne bruge.

Der gik rent faktisk lang tid - før jeg fattede meningen - uden jeg dog kunne forklare den, og da jeg havde/fik med kunder at gøre, var en forklaring rar at have. Endelig fandt jeg en forklaring, jeg kunne lide og forstå:

- Hvis det er der, og du kan se det, er det virkeligt.
- Hvis det er der, og du *ikke* kan se det, er det gennemsigtigt .
- Hvis det *ikke* er der, og du kan se det, er det **virtuelt**.
- Hvis det ikke er der, og du ikke kan se det, er det væk.

IBM begyndte at eksperimentere med virtuel storage allerede i /360-tiden. Man tog en /360 model 40, det var en af de mindre, og ændrede den, så den kunne køre med virtuel storage. Sådant en ændring krævede nogle ekstra registre til at rumme adresseoplysninger, og en enhed, der kunne lave virtuelle adresser om til reelle - en såkaldt DAT (**D**ynamic **A**dress **T**ranslation). Hvis ikke maskinen skulle blive meget langsom, skulle denne enhed naturligvis være særdeles hurtig og adresse omregningen skulle helst foregå samtidig med noget andet.

Da man nu havde bygget maskinen, ville man naturligvis gerne prøve at køre med den. Det var jo i sidste ende det, det drejede sig om.

For at køre på en avanceret maskine, skal man have et operativsystem, og hvis maskinen skal udnytte virtuel storage, skal operativsystemet naturligvis kende til virtuel storage. Der var ingen af IBM's operativsystemer, der havde det mindste kendskab til virtuel storage, og som tidligere antydte, er det ikke trivielt at modificere/kode styresystemer.

Man valgte derfor at skrive et nyt, meget enkelt (primitivt), styresystem. Det skulle ikke være et egentligt operativsystem, men blot en overbygning, der skulle kunne håndtere denne, lidt besværlige, virtuelle storage. Det eneste dette system skulle, var at tage sig af 'pagefault' altså de situationer, hvor et program refererede noget storage, der lå på disk, trafikken med pages ind og ud af maskinhukommelsen og regnskabet med pages i hukommelsen og på disk. Det var ret enkelt sammenlignet med de andre styresystemer, der jo skulle kunne håndtere input/output enheder, dataset og alt sådan noget. Under dette nye system lod man så et standard (helt umodificeret) operativsystem klare resten af styrefunktionerne. Da man havde fået dette system til at virke, erkendte man, at man nu havde fået en maskine til at køre med to vidt forskellige styresystemer samtidig. Det nye system var naturligvis bygget op med alle informationer i kontrolblokke, så da man prøvede at dublere

disse kontrolblokke, opnåede man at kunne køre flere operativsystemer i samme maskine.

Nu er vi tæt ved noget, der lyder umuligt. Et operativsystem, er en kontrolprogram, hvis opgave det er at fordele og udnytte ressourcerne i en maskine. Hvordan kan det så være muligt at have flere operativ systemer i samme maskine???

Svaret er enkelt, det nye styresystem skabte virtuelle maskiner. Hver eneste bruger på dette system, fik stillet 'noget' til rådighed, der svarede til en virkelig maskine, men den var der altså ikke - se definitionen ovenfor på virtuel.

Det gik naturligvis ikke helt så hurtigt, men man fik altså bygget et system, der blev annonceret under navnet CP-67. Jeg går ud fra det stod for **C**ontrol **P**rogram 1967. Til CP-67 annoncerede man et lille primitivt interaktivt end-user-system kaldet CMS (**C**ambridge **M**onitor **S**ystem). CMS var et single user timesharing system og derfor også meget enkelt, det er jo lettere at dele tiden, hvis der kun er en enkelt bruger at dele den imellem.

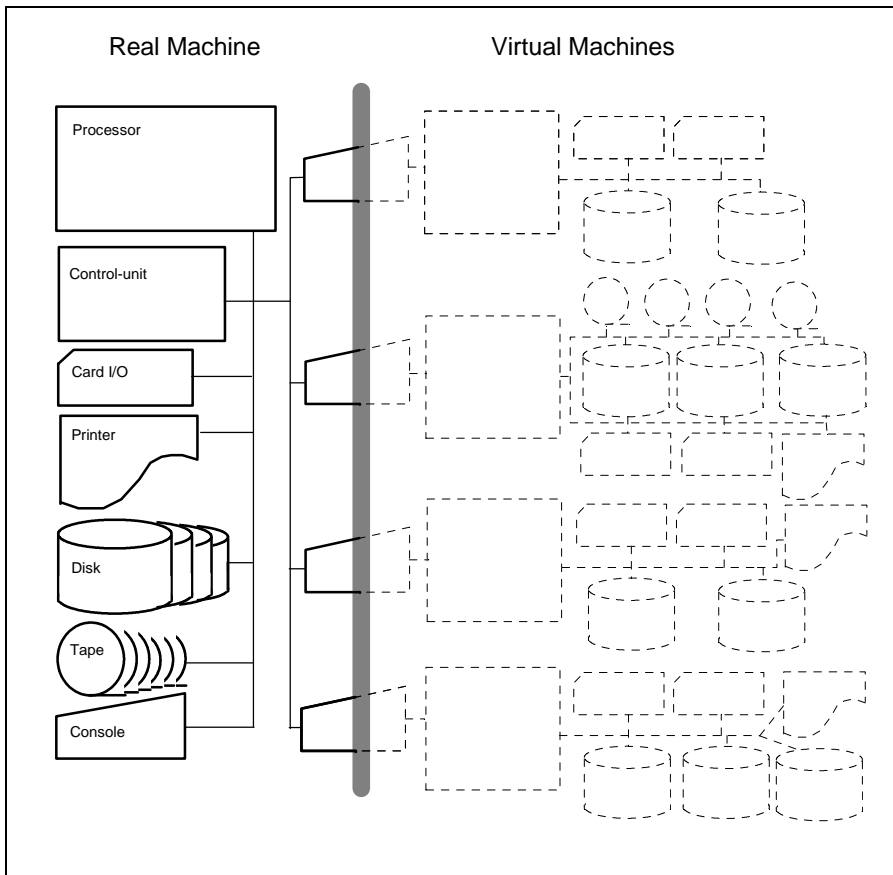
CP-67 og det dertil hørende CMS system fik jeg aldrig stiftet bekendtskab med. Jeg hørte om det, og syntes - det husker jeg stadig - at det lød som ren magi. Magi er noget jeg *aldrig* har troet på, men må indrømme at jeg har ofte haft behov for at *minde* mig selv om at jeg *ikke* gør det.



Skærbillede af en VM-terminal

På den følgende side, har jeg forsøgt at illustrere VM's virkemåde:

Skitse, der viser sammenhængen mellem den reelle og de virtuelle maskiner.



På venstre side af den lodrette kraftige grå linie, er den virkelige maskine vist, på den anden side ses de virtuelle maskiner. Terminalerne på den virkelige maskine bliver konsoller på de virtuelle maskiner, de er derfor tegnet på grænsefladen - den lodrette grå linie. Dette betyder at alle brugere - uden undtagelse - får en virtuel maskine stillet til rådighed, og alle skal derfor have et operativsystem. Dette operativsystem kan være DOS, OS/MFT, OS/MVT, VS1, MVS, VM/370 eller evt. noget hjemmegjort, men det mest populære operativsystem i VM-verdenen er CMS (under VM, står CMS for **C**onversational **M**onitor **S**ystem), der er et end-user system, der gør det muligt at anvende en virtuel maskine næsten som man i dag anvender en PC. Da mange af ressourcerne er delte, skal sammenligningen nok være med en PC, der er knyttet til et netværk (LAN). Muligheden for at køre 'gæste

operativsystemer' under VM blev meget populær måske især hos system-programmører, da det betød at det meste af systemvedligeholdelses arbejdet og aftestning kunne laves om dagen evt. I en virtuel maskine, og ikke som før, om natten eller værre endnu i weekends.

Tapes og diske bliver efter behov overdraget til de forskellige virtuelle maskiner, der kan have konfigurationer efter behov. Med diske er der den ekstra specialitet at en disk kan opdeles i små diske (derfor kaldet minidiske), der af de virtuelle maskiner opfattes som virkelige diske (bare mindre), derfor kan der være flere diske på de virtuelle maskiner, end der er på den virkelige. Disse diske har den fordel at de kan deles mellem flere virtuelle maskiner. Minidiske gør det også muligt at alle brugere kan få en eller flere private diske, der dog efter behov kan benyttes af andre brugere - naturligvis under fuld sikkerhed. Tapes kan naturligvis *ikke* deles, og hulkortudstyr og printere deles som sædvanligt ved hjælp af et SPOOL-system. Jeg kan stadig tydeligt huske at jeg ikke var helt tilfreds med VM's SPOOL-system - HASP var bedre og mere elegant og langt lettere at modificere!!!

I sommeren 1973 var der to systemkonsulenter, der var ved at forberede en stor demonstration af VM/370 for et universitetscenter. Min chef syntes at VM/370 lød spændende fordi det indebar så mange interessante muligheder også for DCS, så jeg blev sat på opgaven for at også vi (DCS) kunne få erfaringer med dette revolutionerende system. Det ubehagelige var, at vi var nødt til at lave *alt* arbejdet om natten. Dette var måske paradoksalt, fordi VM/370 netop var et system, der gjorde det muligt at lave alting samtidig med at andet kører. Jeg må dog erkende at stabiliteten af systemet på *det* tidspunkt måske (*bestemt* er nok mere korrekt) ikke var god nok til DCS's produktionskørsler, så naturligvis kunne vi endnu ikke bruge systemet i et produktionsmiljø.

Vi gik i gang med at lave et system. Under dette system skulle vi køre en DOS maskine, der skulle køre almindelige DOS programmer, en anden DOS maskine, der skulle køre APL, en tredje virtuel maskine skulle køre VM-programmet RSCS (**R**emote **S**pooling **C**ontrol **S**ystem), der skulle bruges til remote job entry. Vi fik stablet demonstrationen på benene, det gik fint, så fint at den blev afholdt adskillige gange og altid med succes. Det var igen lidt af en pionertid Vore terminaler var skrivemaskinebaserede, med den skrivemaskine der skrev med en 'golfkugle'. Jeg husker endnu at en af gutterne efter en vellykket demonstration pludselig gik hen til en skærmtterminal og tændte den, hvorefter han stolt viste at den nu havde et VM logo. Joh det var lidt af en sensation, dengang var skærmtterminaler meget sjældne, men heldigvis blev det jo sådan nogle vi fremover skulle arbejde med. Sommeren 73 står stadig i min erindring som en periode, hvor der kun blev arbejdet om natten, men også som indgangen til en ny og meget vigtig del af min tilværelse, og jeg opnåede da også en mindre kontant godtgørelse som belønning for min indsats. Det

var ganske rart, for vi flyttede fra Roskilde til Kokkedal i efteråret 1973 og kunne derfor godt bruge lidt ekstra - jeg mener at huske at pengene blev omsat til nogle altanmøbler. Langt vigtigere var det dog at VM blev den vigtigste og mest spændende del af min karriere i IBM.

Arbejdet med VM var vældig interessant. Jeg husker en mørk nat, hvor jeg sad sammen med en APL specialist og rodede dybt nede i systemet - vi rettede blandt andet på nogle PSW's. *Da* gik det op for mig at vi havde fuld rådighed over vor del af maskinen (altså den virtuelle) uden at kunne genere de andre der samtidig arbejdede på andre dele af projektet - det var lige så morsomt som at køre på 1620, fordi vi kunne gøre *helt* hvad vi ville uden at påvirke det øvrige system. Det var faktisk en slags personlig computer, måske i virkeligheden den første rigtige PC. Mærkelig nok er det faktisk muligt at blive begejstret også selv om det er en meget mørk og sen nat og man derfor er meget træt.

VM/370 blev efterhånden mere stabil, så 'vi' fik overbevist ledelsen om at det var en god idé at køre systemet i regelmæssig produktion for at også andre kunne få erfaringer med det. Vi troede naturligvis på at dette smarte system måtte have en stor fremtid, hvad der jo heldigvis kom til at holde stik. Det resulterede i at vi fik adgang til en relativ lille maskine om eftermiddagen fra kl. 13.00 til 16.00. Det var ikke lang tid, men det gav os mulighed for at lære systemet og dets muligheder bedre at kende. Det var dengang der *ikke* stod en terminal på alle skriveborde, hvorfor skulle der det, der var jo ingen systemer til at anvende terminalerne endnu, men vi havde dog et rum med en tre-fire skærmterminaler, så *der* brugte jeg mine eftermiddage. Efterhånden begyndte en fyr at dukke op temmelig regelmæssig og kiggede mig over skulderen. Han stillede meget interesserede spørgsmål om, hvad der foregik. Vi kan jo kalde ham Jon (det hedder han *næsten*). Jeg kendte ham godt. Han var polytekniker og havde tilbragt sin praktiktid hos IBM, delvis med at irritere sine omgivelser. Han kastede sig med stor iver ud diskussioner om mangt og meget, og det irriterende var, at han altid havde/fik ret. Efter eksamen var han så blevet fast ansat som systemkonsulent, og han var altså nu ved at kaste sin kærlighed på VM - ligesom jeg.

I begyndelsen kunne jeg naturligvis lære ham meget, jeg havde jo haft fingrene langt nede i systemet et godt stykke tid, men det viste sig hurtigt at han var (og stadig er) langt den klogeste. Jeg overdriver ikke, når jeg hævder at vi sammen voksede op - med VM. Det blev et langt og frugtbart samarbejde - måske især for mig. I IBM Danmark lærte man hurtigt at når det drejede sig om VM var det Jon - subsidiært Preben man skulle kontakte.

Jeg husker en eftermiddag, klokken nærmede sig fire, så det var tid at lukke VM, så maskinen kunne bruges til andre ting. Operatøren havde været i telefonen og fortalt os at vi skulle se at få pakket sammen. Jon loggede så på en privilegeret maskine (det var jo 'vores' system), og afbrød forbindelsen til operatørens konsol og smed

ham af systemet. Nu havde vi kontrollen - troede vi. Ca. 15 sek. senere ringede en *meget* ophidset operatør og fortalte at hvis vi ikke omgående, ville han trykke på IPL-knappen. IPL-knappen er den, man benytter, når man skal starte et nyt operativsystem. IPL-knappen kunne vi ikke sætte ud af funktion - det var stadig operatøren der havde kontrollen, egentlig nok ret betryggende, specielt når systemet skal bruges til produktion.

I begyndelsen blev VM mest benyttet af systemkonsulenter, der ville have erfaring med systemet. Det var altså noget af en pioner tid, på en måde mindede det om *den* tid med 1620, jeg havde været så glad for - altså igen en fornemmelse af 'de gode gamle dage'. Den slags arbejde er altid morsomt, og det passer vist godt til mit temperament. Vi hjalp hinanden med at løse de problemer, der dukkede op - og problemer var der sandelig ikke mangel på, men det er sikkert en god måde at lære på, vi blev i hvertfald dygtige efter hånden.

Tekstbehandling i VM blev klaret af et program, der hed SCRIPT (det må ikke forveksles med et script, der vel nærmest er en form for et program). SCRIPT bliver stadig brugt på værts computere, dog i en noget mere sofistikeret form end det vi havde den gang. Script var lavet af EDB folk (VM'ere) til deres eget brug, for at de kunne dokumentere noget af deres arbejde. Forskellen på SCRIPT og moderne tekstbehandlings programmer er at man i SCRIPT indsatte kommandoer - TAGS - i teksten for at fortælle SCRIPT, hvordan teksten skulle formateres. Datidens skærme var jo ikke grafiske, så WYSIWYG var jo ganske utænkeligt. Vi brugte naturligvis også SCRIPT. Jeg husker stadig hvor begejstrede vi alle var, da vi fik en printer, der *altid* var forsynet med en printkæde med store og små bogstaver og A4 papir (næsten). Nu kunne vi lave nogle flotte skrivelser og nydelig dokumentation af vort arbejde. Jeg tror stadig jeg har et par af disse lister. Jeg ved ikke om det er listernes alder, men jeg synes ikke mere de er ret pæne. Laser-printere eller min egen beskedne Ink-Jet printer skriver altså en hel del flottere, men måske ikke helt så hurtigt, og et moderne tekstbehandlings program kan en hel del mere end SCRIPT kunne i de dage.

Efter et stykke tid - jeg mener dog det gik ret hurtigt - blev VM så populært at vi fik en maskine der udelukkende kørte VM. Nu kunne vi altså køre VM hele dagen. Noget tilsvarende skete i de andre landes datacentre, og så var der én, der fik den strålende idé, at vi skulle arbejde sammen. Det viste sig at være alle tiders idé. Der blev oprettet noget man kaldte 'VM Focal Point'. Det var en 'klub' af system-programmører fra de datacenter installationer, der kørte VM. Vi blev inviteret på en tur til Paris for få idéen præsenteret.

Nu bliver Paris jo normalt anset for at være en yderst spændende by, men jeg må erkende at det, at møde ligesindede fra mange lande var langt mere spændende. Da VM Focal Point var på sit højeste, mødtes vi et par gange om året. Stockholm,

Paris og Rom, var byer, som jeg fik besøgt ved disse lejligheder. Stockholm var måske valgt som mødested fordi den franskmænd, der var projektleder, var gift med en svensker. Det var stort set de samme systemprogrammører, der kom til disse møder, og jeg må indrømme, at der er flere af deltagerne som jeg stadig regner for mine venner - selv om jeg ikke har haft kontakt med dem i mange år, og sikkert aldrig får chance for at gense dem. Møder i Rom startede normalt en mandag efter frokost. Så kunne deltagerne nå at komme med en morgenflyver. Bare ikke hvis man kom fra Skandinavien (med SAS). Så vi plejede at tage af sted om søndagen, og så naturligvis ikke med aftenflyveren. I en by som Rom gælder det om at have så megen tid som muligt til turist brug - Rom er *meget* spændende, den mest spændende by jeg har oplevet. En søndag aften havde Ragner (han kom fra Norge) set et skakbræt, som han var faldet for, han mente det ville pynte i hans dagligstue. Så næste dag vandrede vi til butikken, og brættet blev købt. Det var af marmor og brikkerne var store tinfigurer. Det gik lidt ud over glæden ved at spadser i Rom, at vi skulle slæbe på alt det. Marmor og tin vejer ret meget, og det var som om, det blev tungere og tungere. De næste mange gange, vi mødtes, talte vi om dette skakbræt, men Ragner påstod energisk at det pyntede gevaldigt i hans stue.

Da vi kom tilbage til hotellet med skakspillet, fandt Ragnar ud af at Thor (han er fra Sverige) kunne spille skak. Det blev straks til en gevaldig dyst. Vi var i godt humør, og vist nok ikke helt ædru, men da det gik op for Thor at han havde tabt - og til en nordmand, tror jeg nok, han kom lidt mere til sig selv. I hvert faldt vandt han alle de følgende spil - helt uden problemer - og Ragnar, der havde været så stolt over sin sejr (over en svensker), blev mere og mere frustreret. Alt i alt en hyggelig aften - jeg var jo kun tilskuer.

Det var typisk for disse internationale møder at vi skandinaver som oftest søgte sammen. Jeg husker en dag vi sad på en restaurant og fik noget frokost. Der var Risto fra Finland, Thor fra Stockholm, Ragner fra Oslo og så lille mig fra København. Vi sad og sludrede om løst og fast, de nordiske sprog ligner jo heldigvis hinanden så meget at det ikke er noget problem (ja, der er vist nok et lille problem med finsk). Så dukkede denne amerikaner op og sagde: "Når I nu taler sammen, taler I så norsk, svensk eller dansk??" Han så synligt forvirret ud, da vi alle svarede: "Ja!!". Jeg må dog erkende at Risto havde et problem, det er som om finner har meget svært ved at forstå dansk - egentlig mærkeligt, da en dansker har lettere ved at forstå en svensk talende end en svensker.

Dokumentation er ikke noget 'rigtige' EDB-folk er vilde med. Men det skulle jo gøres. Jeg tror nok det var mig, der tog initiativet (jeg er ikke sikker, det er jo så længe siden), men det var i hvert fald mig der udviklede en række procedurer, der gjorde det muligt at afdelingen fik dokumenteret sit arbejde. Hver systemprogrammør havde ansvaret for at dokumentere sin del. De enkelte afsnit blev så placeret på en måde så jeg kunne udskrive hele værket - uafhængig af de enkelte 'forfattere'. Det var ret fornemt med rettelses mærker i marginen (de der lodrette streger), så

man hurtigt kunne se, hvad der var nyt. Jeg husker slet ikke hvordan det blev gjort, men jeg kan da huske at det var betragtelige mængder af papir, der blev brugt, når en udgave blev udskrevet i et passende antal eksemplarer, og selve programmerne var vist også temmelig sofistikerede - jeg mener at huske at der var ret mange der var imponerede, også nogle af dem, hvis mening jeg havde respekt for. Det jeg husker bedst er dog de store mængder papir, der kom ud af det. Hvad der blev af dette system kan jeg imidlertid ikke huske.

Af sikkerhedsmæssige årsager, var det *strengt* forbudt at forlade sin plads med sin skærm online. Jeg fandt hurtigt ud af at det var for besværligt og tidsrøvende at lukke sin VM session hver gang man lige skulle et eller andet. Jeg lavede så - ved hjælp af den uundværlige XEDIT et billede, der til forveksling lignede det, der standard sad på skærmen, når den blev tændt. Jeg fik oven i købet billedet opdateret på samme måde som standard billedet, når man trykkede på en eller anden tast. Dette billede kunne kun fjernes, hvis man kendte et password (nøgleord). Det virkede fint, men en dag kom der besked fra sikkerheds folkene om at man *ikke* måtte bruge den slags små kneb, de var *alt* for usikre. Jeg blev gal i skralden, jeg synes ikke de der sikkerhedspampere kunne tillade sig at kritisere *mit* arbejde. Det fortalte jeg dem så. Jeg fik en lang historie om at de ikke kunne overkomme at teste alle mulige menneskers mere eller mindre umulige forsøg på at låse skærmen på en forsvarlig måde. Meen, han ville da gerne gøre en undtagelse i mit tilfælde, jeg var jo trods alt VM specialisten, så man havde lidt respekt for min mening. Jeg sendte ham så rutinen, og fik naturligvis det forventede svar. Han kunne ikke finde nogen fejl i min rutine. Selvfølgelig ikke, jeg havde da lavet den efter alle kunstens regler - og et par stykker til, så der var (naturligvis) ikke nogen mulighed for at snyde min rutine. Jeg har altid været *meget* motiveret for sikkerhed, så jeg blev da ved med at bruge den så længe jeg kørte VM.

En af mine opgaver var at holde øje med, hvor effektivt maskinen kørte. Der var forskellige måleværktøjer, der kunne hjælpe mig, så jeg (forhåbentlig) blev i stand til at dreje på de rigtige knapper for at få maskinen til at køre optimalt. Når jeg nu var i gang, var det nærliggende at måle hvor belastet maskinen var. Det gjorde jeg over en ret lang tid, og så begyndte jeg at interessere mig lidt for statistik - naturligvis kun på amatørbasis, men jeg regnede på udviklingen i maskinforbrug og kunne se at det voksede hele tiden. Jeg skrev derpå en rapport til min chef og fortalte ham hvornår jeg mente vi ville nå loftet. Det så ud til at mine beregninger var korrekte, for vi fik performance problemer omtrent på det tidspunkt, jeg forudså. Endnu en lille fjer i hatten.

Det var på den tid, der kom noget nyt og meget spændende: VNET. VNET var begyndt som et telekommunikations netværk mellem IBM's laboratorier. Senere blev IBM kontorer, der lå i nærheden af laboratorierne koblet på, og senere - meget

senere kom *alle* med. VNET var for IBM, hvad internet/e-mail er for hele verden i dag. Ved hjælp af VNET kunne man sende elektronisk post (E-mail, vi kaldte dem VMSG's og senere NOTE's) og filer til de andre på nettet. Man kunne helt klart dele IBM op. Der var dem på nettet - og så de andre. Dem på nettet havde man let og hurtig kontakt med, de andre kunne man kun få fat i pr. telefon eller gennem postvæsenet, det betød jo så at man skulle til at tale engelsk eller endnu værre skrive det på papir, hvilket er meget mere formelt end en elektronisk skrivelse, så de blev nok koblet lidt ud på et sidespor.

En ting jeg husker tydeligt er de modems vi benyttede. Et moderne modem er vel en 15 - 20 cm lang og et par centimeter høj. De modems vi benyttede den gang var vel nærmest på størrelse med en moderne stationær PC. Til gengæld krævede de at kommunikere med et andet modem, der var nøjagtig magen til. Jeg husker desværre ikke hvor hurtigt vi kunne køre, men det foregik alt sammen på faste linier.

I Danmark kom vi ret hurtigt på nettet - takket være Jon. Jon var efterhånden blevet en særdeles højt anset kapacitet og han havde naturligvis kontakter til de *rigtige* steder (laboratorierne), det betød så at han var særdeles aktiv for at få os på nettet. Det viste sig til vort held at vi (IBM Danmark) havde en direkte telefonlinie til Stockholm, der ikke blev brugt, og Stockholm var på nettet takket være laboratoriet på Lidingö. Vi fik også på mystisk vis fat i et modem, blev defineret på nettet (det ordnede Jon og 'vennerne'). Så havde vi de sædvanlige startproblemer. Jeg ved ikke hvorfor sådan noget aldrig virker første gang, men det må være en eller anden ikke beskrevet naturlov. Efter nogle forsøg - det tog flere dage, var der pludselig forbindelse og det har der lige siden været, næsten konstant.

VNET kører hovedsageligt mellem systemer, der kører VM. Man benytter programmet RSCS (**R**emote **S**pooling **C**ontrol **S**ystem) til kommunikationen. Nettet kører næsten ukontrolleret og det er et *fantastisk* værktøj. Specielt dengang, da man ikke var vandt til computer net, som vi er det i dag.

Jeg husker en eftermiddag. Jeg havde i et dokument (formodentlig elektronisk) set at en fyr i Australien havde udviklet nogle full-screen programmer, der gjorde det muligt at få adgang til MVS filer fra CMS. Jeg skrev til ham lige inden jeg gik hjem. Næste morgen da jeg kom på kontoret lå programmerne i VM og ventede. De blev hurtigt installeret og kørte alle ved frokosttid. Det var noget helt andet end postvæsenet - og så kom de jo fra den anden side af kloden!!!

Programmerne var også ganske fornøjelige. Fra CMS havde jeg faktisk bedre og mere elegant adgang til MVS-filer, end man havde fra MVS. MVS systemprogrammørerne gjorde i sandhed store øjne. Men netop den slags var vel en væsentlig årsag til at vi VM'ere holdt (holder) så meget af 'vort' kære VM. På VM kunne vi som regel gøre mindst det samme som man kunne på de andre systemer, men det var gerne meget mere elegant og helt bestemt *meget* hurtigere. Jeg fik meget hurtigt en følelse af, at hvis man én gang havde prøvet at køre VM, var det svært at føle sig tilpas med andre systemer.

VNET voksede hurtigt. Da node nummer 1000 dukkede op, viste det sig, at det var teknisk afdeling i København. Det blev naturligvis fejret, 'deres' VM-mand blev inviteret til et fint møde i Chicago.

VNET lever i øvrigt stadig i bedste velgående, og det er naturligvis i dag koblet sammen med internettet. Jeg har flere gange kommunikeret med *ikke IBM'ere* på internettet via VM.

Også for Focal Point var VNET en herlig opfindelse - altså for os, der *var* på nettet. Vi havde lært hinanden at kende til møderne, og så var det en let sag at holde kontakt over nettet. Vi skrev sammen rigtig tit og det *er* meget lettere at skrive til folk man har mødt, end til helt fremmede, og samtidig er den elektroniske post ikke helt så forpligtende, den bliver f.eks. kun læst af modtageren, medens almindelig post går gennem postafdelingen og diverse chefer før den når til modtageren.

Jeg husker en opdatering af VM, der var nødvendig fordi der var kommet en ny generation af skærmterminaler. Det betød at man måtte ændre (udvide) kontrolblokken, hvor skærmene var definerede (der var brug for flere oplysninger). Koderne, der fortalte at en device var en skærm, blev lavet om. Det havde katastrofale resultater for vore interne full-screen programmer. Pludselig troede de ikke mere at den anvendte CMS konsol *virkelig* var en skærm. Da vi ikke havde kildeprogrammer til disse programmer, var det muligt/nødvendig at lede efter testen på den pågældende kontrolblok i maskinkoden. Den slags er langt fra trivielt - eller sagt på en anden måde - det kræver en ægte 'bit-vender' - altså en *rigtig* systemprogrammør. Men når man så havde fundet fejlen og rettet i et program, var det en sand fryd at sende rettelsen til alle vennerne i Focal Point. Jeg ved ikke *hvor* meget det betød, men der blev sparet rigtig mange mandetimer (dage??) fordi et program nu kun skulle rettes ét sted. Samtidig var de fleste af programmerne værktøjer, der øgede folks produktivitet, så jo hurtigere de kom til at virke over alt, jo hurtigere gjorde de gavn igen. Jeg tror nok, jeg trak min del af læsset, men det var måske fordi, jeg var mere afhængig af de pågældende programmer end de øvrige systemprogrammører. Joh, jeg syntes da også det var vældig sjovt.

På et af disse internationale møder, fik jeg orientering om et program, det hed SMART (Systems Measurement and Analysis in **R**eal Time). Det var et program, der til fulde, levede op til sit navn. Det kunne fortælle om, hvor godt/dårligt VM kørte - medens man kørte. Andre tilsvarende programmer gjorde det ved en analyse efter man havde kørt!!! Naturligvis installerede jeg SMART, da jeg kom hjem, og naturligvis fandt jeg fejl i SMART. Nogle af fejlene lavede jeg rettelser til, så jeg begyndte at kommunikere med forfatteren. Det blev til rigtig mange 'breve' gennem nettet, så efterhånden syntes jeg kendte ham rigtig godt. Han hedder i øvrigt Jensen (han har forlængst forladt IBM), så jeg skrev og spurgte ham om

han havde nogle relationer til Skandinavien. Joh, hans forældre (eller var det bedsteforældre??) var fra Danmark, og han var helt vild efter at se 'Det gamle land', så hvis jeg kendte nogen, der var interesseret i et seminar om SMART, og som var villige til at betale, så ville han være der, lige som et søm... SMART lever i øvrigt i bedste velgående, nu under navnet RTM (**R**ea**T**i**M**e **M**onitor).

En dag var jeg i maskinstuen, da man var ved at starte VM. Det var helt trivielt, men da jeg stod ved konsollen, da VM stillede det traditionelle spørgsmål om hvordan vi ville have starten skulle foregå, skrev jeg 'COLD' hvor jeg skulle have skrevet 'WARM'. Resultatet var at VM fjernede alle sine SPOOL-filer, og dermed hvad der monne være af elektronisk post. Dette skulle normalt ikke være noget problem, da brugerne skulle opbevare deres post alle mulige andre steder end i VM's SPOOL-kø. Der var dog ét problem. Jon havde netop haft en eller anden stor (og vigtig??) kontakt med en masse udlændinge, så han havde ca. 70 noter (altså elektroniske breve) liggende, som han endnu ikke havde fået læst. Han var sur - rigtig sur!! Nu skulle han have fat i alle disse kolleger igen for at bede dem sende posten endnu en gang. Joh, der gik et par dage, hvor han så lidt skævt til mig, jeg tror endog han et par gange gryntede, når vi mødtes (hvad jeg klogelig forsøgte at undgå).

Heldigvis er der retfærdighed her i tilværelsen!!! En lørdag ved frokosttid ringede Jon. Han havde prøvet at lægge en lille (men i følge Jon) vigtig modifikation på VM. Det gav ham et problem, han kunne ikke starte VM. Jon havde ikke adgang til maskinstuen, men vi havde i fællesskab 'opfundet' en metode til at genstarte VM fra en privilegeret bruger. Men VM startede altså ikke. Jon mente der var problemer med modifikationen (hvad der måske også var ?), men *jeg* vidste at teknikerne dagen efter skulle ombygge maskinen, og jeg havde derfor ændret VM definitionerne, så de passede til den nye konfiguration. Jon havde altså genereret en VM, der passede til maskinen, som den ville se ud et par dage efter. Han var *ikke* stolt, men det havde for mig den store fordel at han nu kunne tilgi' mig de manglende SPOOL-filer. Det er nok lettere at tilgive andre, når man så tydeligt - på egen krop - har fornemmet, hvor let man selv kan komme til at lave en alvorlig fejl. Jeg husker ikke om jeg måtte til IBM for at løse problemet, men jeg er helt sikker på maskinen og VM kørte den følgende mandag.

Det er i sandhed menneskelig at fejle. Omgangen med computere har lært mig at hvis nogen påstår de aldrig laver fejl, er de ikke til at stole på!!!!

Alle brugere kører i virtuelle maskiner, men det er naturligvis ikke alle virtuelle maskiner, der kan det samme. VM kommandoerne er inddelt i klasser, og en bruger får så tildelt en eller flere af disse kommandoklasser. Jeg havde naturligvis selv en maskine (der underligt nok hed PREBEN), og da det var der, jeg passede mit job, var det en maskine uden særlige privilegier, det havde to fordele. For det første kunne jeg ikke lave ulykker når jeg dummede mig (det er jo det der med at fejle),

og for det andet vidste jeg at de ting jeg udviklede skulle kunne bruges af alle andre. Imidlertid skete det hyppigt at jeg blev bedt om at gøre et eller andet for en kollega, der krævede privilegerede kommandoer. Det indebar at jeg var nødt til at afbryde min session, og etablere en ny med en privilegeret bruger. Lidt besværligt, så jeg opfandt en service maskine, der kunne *alting*. Så skrev jeg nogle programmer, så jeg kunne dirigere denne service maskine, fra 'PREBEN' og *kun* derfra ved hjælp af meddelelser. Ganske smart (synes jeg selv), jeg bad simpelthen service maskinen om at udføre specielle kommandoer for mig, og det var så specielt at jeg ikke kunne komme til at gøre det ved et uheld. Der var dog én kommando, jeg *ikke* ville tillade, nemlig den der lukker systemet ned. Så jeg lavede en test for denne kommando - uheldigvis havde jeg vendt en hop instruktion forkert, den slags sker desværre, det er nok ham der Murphy igen, eller måske er det endnu et bevis på jeg er menneskelig. Sådan noget skulle naturligvis prøves, så jeg bad min service maskine om at lukke systemet. *Det gjorde den*. Chefoperatøren var ganske ophidset, da han fortalte mig at dersom jeg gjorde det der én gang til, skulle han personligt sørge for at jeg aldrig mere fik specielle privilegier til *noget* VM system - vistnok på hele kloden. Det kunne han naturligvis ikke, men jeg sagde pænt undskyld, og vi forblev (heldigvis) gode venner, det mener jeg da, og jeg var da langt senere med til hans 25 års jubilæum, så på det tidspunkt må han da have tilgivet mig.

IBM har et udmærket system, der skal belønne medarbejderne for kreativitet - nemlig en forslagskasse. Den har jeg brug to gange.

Teknisk afdeling havde en overgang for vane at udsende VM rettelser skriftlig. Det var *så* systemprogrammørens job at overføre disse rettelser til et medie, som maskinen kunne læse. Den slags giver som bekendt rig mulighed for fejl, de er jo også mennesker. Jeg fik så den idé at man forholdsvis let kunne få rettelserne direkte overført til et magnetbånd i stedet for på papir. Det var noget med at teknikerne hentede oplysningerne fra en database, men de kunne kun udskrive dem på papir. Jeg havde i mellemtiden fundet ud af hvordan man kunne dokumentere den session, således at man fik opdateringer i maskinlæsar form og man derved kunne undgå den manuelle indtastning. Jeg brugte altså et kvarter til at skrive en halv A4 side med denne idé. Den blev afvist!! Nå, det må man jo leve med. Et stykke tid senere fik jeg VM rettelser fra teknisk afdeling - på et magnetbånd!!! Det virkede lidt som snyd. Man kan da ikke afvise et forslag, for derefter at bruge det. Jeg styrtede ned til 'chefen' for forslagskassen. Han sad netop med et telex fra England, hvoraf det fremgik at man syntes det var en fremragende idé som man burde støtte. Nogen tid efter kom afdelingschefen med en check på ca. 1800 Kr. - en ganske pæn belønning for et kvarters arbejde (i arbejdstiden). Senere blev jeg overflyttet til en salgsafdeling, og en skønne dag blev jeg kaldt ind til min nye chef, der havde to checks til mig. I alt tror jeg fik næsten 5000 Kr. for denne lille - men åbenbart ikke helt ringe idé.

Denne succes gav mig naturligvis blod på tanden. En dag - i bilen på vej til sommerhuset - fik jeg en anden idé. Det var noget langt mere kompliceret og langt mere værdifuldt for IBM. Jeg husker ingen detaljer, men det var noget med fælles dataadgang fra VM og MVS - under fuld sikkerhed naturligvis. Fælles adgang havde der hele tiden været mulighed for, problemet var at man ikke kunne tildele adgangen på dataset niveau. Dette forslag fik megen ros. En kollega havde kontakt med en IBM'er fra USA, der havde hørt om det. Alle tiders idé. Efter den besked forventede jeg naturligvis er mindre formue. Men nej. Det var et forslag, der havde med mit arbejde at gøre, og et sådant kunne ikke belønnes. Lidt skuffende, men en hit rate på 50% er vel ikke helt dårlig, så jeg besluttede at den ville jeg ikke risikere at ødelægge ved at skrive flere forslag. Én ting finder jeg dog urimelig. Når et forslag bliver belønnet, beskattes det som normal indtægt. Hvis man derimod vinder penge ved at spille, tippe og sådan noget, bliver gevinsten kun beskattet med 15%. Jeg synes man burde belønne kreativitet frem for spillelidenskab.

Det er morsomt at have kontakt med kolleger fra udlandet. I min tid som VM'er har jeg, takket været VNET, haft kontakt med en masse mennesker, som jeg desværre aldrig får at se. Nogle af dem synes man, man kommer til at kende rigtig godt. Der var en fyr i Californien, der havde skrevet et program, der simulerede en HP lommeregner (det var den gang lommeregnerne var meget dyre), en anden fyr i Uithoorn (det er i Holland) havde givet 'lommeregneren' et mere elegant udseende, men denne ændring var lavet som en program modifikation, hvilket betød at hver gang amerikaneren ændrede/rettede sit program, skulle man vente på at hollænderen også lavede sin del, før man kunne få glæde af det. Jeg fik den idé at man kunne kalde amerikanerens program med en edit-makro, med det resultat at ændringerne fik virkning med det samme, da makroen ikke var afhængig af programmet - og omvendt.

For at min idé kunne virke, skulle der kun ændres nogle få linier i det originale program, så jeg skrev til forfatteren og satte ham ind i mine tanker. Han var vildt begejstret, fordi det, med min idé, var muligt at lave 'lommeregneren' med permanent hukommelse og det var noget end ikke HP havde gjort på den tid. Men inden vi var nået så langt, syntes jeg, at vi kendte hinanden rigtig godt, og i sådan en situation, ville det have været morsomt at kunne smutte 'over there' for at sige 'Hi'.

Desværre har IBM aldrig ment at det var i deres interesse at sende mig til Californien.

En dag skete noget mærkelig. VM begyndte at køre langsommere og langsommere. Nu er VM så smart til at fordele ressourcerne mellem brugerne, at man tit kan passe sit arbejde på VM uden man behøver at mærke at der er problemer, så jeg sad og arbejdede uden at mærke noget. Men operatørerne kom altså og fortalte at der var noget rivende galt. Det var let at se, hvad det var. Det område VM benyttede til at

opbevare sine kontrolblokke, voksede og voksede. Det resulterede naturligvis i at der blev mindre og mindre plads til de kørende programmer. Problemet - altså det rigtige - var at man havde bedt VM om at benytte en kontrolenhed til noget andet, end det, den var beregnet til. VM prøvede hele tiden at 'tale' til kontrolenheden, og det gav en fejl. En fejl registreres først i maskinens hukommelse i en kontrolblok, der så bliver skrevet ned på en disk, når der er tid til det. Fejlene kom desværre bare hurtigere, end man kunne skrive på disken, derfor hobede disse fejl kontrolblokke sig op. Da man afbrød kontrolenheden, gik der ikke så lang tid før VM igen forøjeblikket kørte videre for fuld fart.

Efter et stykke tid var der for lidt diskplads - det er naturligvis noget der ofte hændte, men denne gang var der ikke plads til flere diske i maskinstuen! Det blev så besluttet at udvide maskinstuen, ved at inddrage nogle kælderrum. Her kunne man så placere diske og andet udstyr, der ikke krævede operatør-assistance. Man borede altså nogle gevaldige huller i betonvæggene - det er ikke nogen lille sag. Gennem disse huller kunne man så trække interface kablerne (det er dem, der knytter maskinens 'kasser' sammen), så de nye disk kontrolenheder og de nye diske kunne stå i kælderen.. For at flytte, med mindst mulig nedbrud, kopierede man backuptapene for de 'gamle' diske ned på de nye diske, som man derefter tog i brug. De gamle diske skulle naturligvis flyttes senere. Alt i alt en udmærket måde at gøre tingene på, altså under forudsætning af at man hele tiden ved, hvad man gør.

En aften, lige efter jeg var kommet hjem, ringede telefonen. En kollega beklagede sig over at hans minidisk pludselig så ud, ganske som den gjorde et par uger tidligere. Nu må jeg nævne at det ikke var *helt* usædvanligt at der kom de mest besynderlige klager over mærkelige ting, så jeg prøvede at berolige ham - én gang er som bekendt ingen gang. Der kom imidlertid et par lignende opringninger samme aften, så efterhånden gik det op for mig at det *var* noget *helt* galt.

Så er det man må til at fundere over, hvad der kunne være sket. Det er helt bestemt en af de situationer, hvor man ikke har en chance, hvis man ikke kender systemet (altså VM) og sine medmennesker temmelig grundigt. Det jeg mener *er*, at den slags kræver en virkelig specialist, vi var sikkert kun en to tre stykker i IBM, der ville kunne løse dette problem.

Når man starter VM, ser den på alle de diske, den kan finde, i adresse rækkefølge. En disk har en elektronisk etiket altså et navn der nødvendigvis skal være entydigt, og hvis der findes flere diske med samme navne, bruges den først fundne - altså den med laveste adresse.

Dagens problem var at man havde flyttet de oprindelige diske ned i kælderen og monteret dem på systemet med lavere adresser end de nye, der havde erstattet dem. Da de 'gamle' diske så ud, som de altid havde set ud, havde de naturligvis *det* navn, de hele tiden havde haft, og de blev derfor accepteret, i stedet for de nye og altså med et indhold, der var et par uger gammelt. De gamle diske skulle naturligvis have været slettet eller initieret med nye 'navne', før de blev taget i brug, men det havde

man altså glemt. Selv noget så enkelt som flytning af diske (eller måske også af andre enheder) kræver altså omhyggelig planlægning med deltagelse af folk, der ved hvad de gør.

Jeg ringede til operatøren, der kunne bekræfte at de 'gamle' diske rent faktisk var flyttet ned i kælderen og koblet på systemet, så min teori viste sig altså at være rigtig.

Det var måske ikke den helt store sejr, men det er rart at finde teoretisk løsningen på et yderst alvorligt problem, og *så* få bekræftet at man rent faktisk har ret - det er altid rart når teori og praksis stemmer overens. Det er sandelig også en god ting at slippe for at styrte ud af døren, fordi alting ser ud til at være forkert, der var jo en enkelt sag for operatøren at slukke for de 'forkerte' diske. Når man endelig *er* kommet hjem til sin familie, vil man gerne have lov til at blive der et stykke tid. Joh, jeg var rigtig godt tilfreds med mig selv den dag, alt i alt var en af de dage, der styrkede min selvtilid. At der så var en eller flere af mine kolleger, der burde have meget røde ører, var jo ikke mit problem.

Jeg havde en herlig tid med VM. Jeg kom rigtig godt ind i det, og fik efterhånden et godt navn i den forbindelse - jeg var dog hele tiden overskygget af Jon, men han var til gengæld altid god at ha' ved hånden. Dygtighed og hjælpsomhed er to *meget* gode kvaliteter, og Jon har dem begge:

- ♦ En eftermiddag døde VM pludselig. Det var ikke *så* usædvanligt, og den genstartede sig selv, som den havde for vane. Kort tid efter døde den igen. Denne gang kunne den ikke genstarte sig selv, da dens IPL disk (altså den disk den skulle startes fra) var ødelagt. Mærkeligt, men kuren var selvsagt at genopbygge disken ved hjælp af en back-up tape. Dette var imidlertid ikke særligt populært, da vor VM maskine delte kanaler med vore MVS maskiner, og da restore programmet kørte med nogle meget lange kanal-programmer, der monopoliserede kanalen, fik MVS ikke rigtig lov til at gøre noget. Efter genopbygning af disken prøvede vi igen, ned op ned og ødelagt disk. Dette var naturligvis ikke særlig rart. VM var meget flink til at dokumentere sine problemer, men dokumentationen blev naturligvis lagt på en disk, og disken kunne kun bruges, når man havde et VM system. Jon var naturligvis en bruger, og blev som sådan generet af at VM ikke kørte. Da han via telefon, var sat ind i problemet, dukkede han op med en programliste under armen.

Vi reparererede disken endnu en gang, og under opstarten rettede Jon 'lidt' i maskinens hukommelse, med det resultat at VM nu skrev sit dump på en linieskriver i stedet for på disk.

Det tog lang tid, sådan en hukommelsesudskrift er et stort, meget stort dokument, men nu kunne vi omsider begynde at lede efter fejlen. Vi fandt hurtigt ud af, hvor det gik galt, det var et ganske lille program-loop, der så

ud til at være helt i orden. Det var igen Jon, der indså at det udelukkende bestod af binære instruktioner. Sådan nogle er meget hurtige (naturligvis fordi alle computere dybt nede på bunden er binære af natur), så problemet viste sig at være at et betinget hop, fandt sted, før betingelsen var sat. Derefter konstruerede Jon et lille program - direkte i maskinens hukommelse - der hele tiden viste fejlen. Nu havde teknikeren ingen chance for at smutte uden om. Jeg tvivler på denne fejl var blevet rettet uden Jon's medvirken, ingen af teknikernes testprogrammer ville være i stand til at finde sådan en fejl.

- ♦ Teknikerne havde brugt en weekend til at modificere vor VM maskine. Man havde ændret i mikrokoden, så den kunne køre TSO/E (hvad det så er for noget). Chefoperatøren havde sidst på søndagen checket at maskinen stadig kunne køre og han havde også startet sin egen virtuelle maskine. Alting så ud til at være i orden. Det var det bestemt ikke!!! Jeg skulle på et kursus om mandagen. Heldigvis var det i huset. Der blev hurtigt sendt bud efter mig. VM opførte sig højst ejendommeligt. Det var, som om man havde adgang til andres virtuelle hukommelse - noget helt uhørt og bestemt ikke i overensstemmelse med VM's sædvanlige sikkerheds standard. Da jeg vidste at VM havde været meget stabil i en lang periode, var jeg helt sikker på at problemet var hardware - altså maskinen. Det sagde jeg til teknikerne. Jeg fik at vide at de havde kørt deres testprogrammer, og at de viste at alting var i orden, så det *var* et VM problem. Der blev tilkaldt et par selvhøjtidelige hardware specialister, der begyndte at diskutere. Så syntes jeg det var på tide at gøre noget effektivt, brugerne ville jo gerne have VM til at køre, så jeg sneg mig ind på mit kontor og ringede hjem til Jon. Jon er *ikke* A-menneske, snarere Z. Så jeg kunne tydeligt høre at jeg havde vækket ham. Han har hemmeligt nummer for ikke at blive forstyrret af alle mulige og især umulige kolleger, så jeg var ikke helt stolt, det er jo en tillidserklæring, når man bliver betroet et hemmeligt nummer. Da jeg havde sat ham ind i problemet, mumlede han søvndrukken: 'Sig til dem at der er noget galt med 'STO stack'. Jeg havde dengang ingen anelse om hvad en 'STO stack' var, men jeg gik ind i rummet, hvor diskussionen fandt sted, og sagde, hvad jeg havde fået besked på. Der blev uhyggelig stille. Hvordan kunne en dum systemprogrammør tillade sig at belære tekniske specialister om sådan noget???? Jon var naturligvis sprunget ud af sengen for at komme til assistance. Han kom ind af døren netop som teknikeren kom i tanke om at han havde glemt at skifte disketten med mikroprogrammerne ud. Det resulterede i at maskinen troede at der var flere registre i STO stack end der rent faktisk var. STO stack er et antal registre, der hver indeholder et ankerpunkt til en brugers virtuelle hukommelse. Der er flere, for at det skal gå hurtigere at skifte fra bruger til bruger. Jon havde altså ret. Netop vækket

af sin søvn, satte han direkte fingeren på det ømme punkt. Se *det* er en ægte guru. Han har godt nok senere påstået at han i forvejen var blevet advaret om at maskinændringen havde noget med STO stacken at gøre, så ud over at være bragende dygtig, kan han altså også være passende beskeden.

Ovenstående eksempler viser forskellen på en specialist - altså sådan én som mig - og en virkelig guru - det er sådan én som Jon. Jeg må erkende at jeg er stolt over at være den, der i sin tid gjorde ham interesseret i VM, og altså på en måde givet ham hans VM grunduddannelse og jeg er meget glad for at have arbejdet meget sammen med ham. Man kan ikke skrive/tale om VM i Danmark uden at nævne Jon's navn adskillige gange, så der kommer sikkert meget mere om Jon senere.

Det at VM og MVS delte kanaler havde flere fordele. Det betød blandt andet at VM kunne få adgang til MVS diskene. Adgangen blev naturligvis styret af systemet, så der var ingen sikkerhedsmæssige problemer. Men vi systemprogrammører kunne naturligvis *alting*. CMS havde support til at læse MVS diske, så hvis man kunne få adgang til en disk, kunne man også læse den. Til tider mere elegant end MVS, men alting var jo *lidt* bedre under VM. En kreativ MVS systemprogrammør fandt ud af at han ville prøve at skrive i et MVS dataset fra VM. Han fik defineret en minidisk, med læse/skrive adgang, der var placeret oven i et MVS dataset, og da han var en af de virkelig skrappe drenge, vidste han alt om hvordan sådan et dataset var opbygget. Han skrev så et assembler program, der gjorde som det skulle, og sandelig om det ikke lykkedes. Han fik skrevet data, og MVS kunne læse dem. Der var næsten ingen grænser for hans stolthed - indtil man ved ugens udgang skulle lave den rutinemæssige back-up af disken. Den gik ikke, slet ikke. Det var noget med disk records, der indeholder en adresse, der er relativ til diskens start, men da VM havde skrevet denne record, var den pågældende adresse relativ til minidiskens start, og det kunne MVS back-up program ikke finde ud af - så denne disk kunne aldrig mere håndteres af back-up programmet. Han gik virkelig i tænkeboksen for at løse problemet, men på trods af at han var særdeles vidende, fandt han ingen løsning på dette pudsige problem.

Jon drog så på assignment i England, jeg husker ikke om det var to eller tre år. I den periode blev VM/370 til VM/SP (SP står for **S**ystem **P**roduct). Det betød at der kom en ny standard for elektronisk post og fil-udveksling, men det vigtigste - set fra brugeren - var nok at EXEC blev erstattet af EXEC2, der kom XEDIT og i en senere release blev EXEC2 suppleret med REXX.

Jeg husker Jon beklagede sig over at den engelske systemprogrammør i starten ikke ville give ham de samme VM privilegier, som han var vant til at have i København. Denne systemprogrammør kendte jo ikke Jon, så han var ikke klar over hvad han gik glip af, men det blev han vistnok efterhånden, og så blev Jon glad igen.

Når alle VM'ere havde deres egen (godt nok 'kun' virtuelle) maskine, var det meget kreativt miljø, hvor man kunne gøre alt, hvad man kunne drømme om. Derfor fik folk de bedste muligheder for at udfolde og udvikle deres idéer. Der havde f.eks. været en masse tekst-editorer, altså det program man f.eks. bruger, når man skal skrive programmer. Når en ny editor blev opfundet, indeholdt den angiveligt det meste af det, de tidligere kunne, plus selvfølgelig noget nyt. Det var egentlig ret forvirrende, men man besluttede så at lave en ny editor, der skulle indeholde alt, hvad de tidligere havde og så igen noget mere. Det lykkedes. Produktet kom til at hedde XEDIT. Navnet kom måske af eXtended **EDIT**or, men onde tunger påstod at fordi forfatteren hed Xavier de Lambertier, var det **X**avier's **EDIT**or. XEDIT bliver vist stadig regnet for en af de bedste editorer, der er skrevet bl.a. fordi den giver mulighed for at man kan konstruere sine egne kommandoer ved hjælp af EXEC2 eller REXX. Den første XEDIT, der kom til landet, var en pre-udgave, der kunne køre på VM/370. Det var Jon, der sendte den til mig - elektronisk. Jeg gik straks i gang med at tilpasse den til mine behov. På det tidspunkt kørte jeg også en del TSO, så jeg fik XEDIT til at minde lidt om TSO's ISPF/PDF editor, så var det mere bekvemt for mig at skifte mellem disse miljøer. Da jeg var den første bruger, kom jeg naturligtvis til at påvirke de øvrige, så der er masser af IBM'ere og IBM's VM-kunder, der stadig kører med en XEDIT profil, der får XEDIT til at ligne ISPF editoren. Det er sjovt at have været med til at påvirke 'verden' på denne måde. Jeg benytter i øvrigt stadig en XEDIT 'clone' ved navn KEDIT på min PC, og jeg har *stor* glæde af den, da mine XEDIT macros fra VM næsten alle kan køre under KEDIT - stort set uændrede.

EXEC2 og REXX er en helt anden historie. Interaktive miljøer, som VM, har næsten altid en mulighed for at skrive kommando-filer. En kommandofil, eller som det hedder på VM'sk en EXEC, er oprindeligt en samling system kommandoer, der er samlet i en fil, der kan bruges som en ny kommando. Altså en mulighed for at udstede en række kommandoer på en bekvem måde. Man *så* i VM hurtigt et behov for at kunne overføre parametre til sådan en exec, og så var det jo også rart med lidt logik - alting skulle jo - som sædvanlig - helst være bedre under VM. Nu var det så muligt at skrive små applikationer, og skønheden ved at gøre det på denne måde var (er) at man ikke skal oversætte til maskinsprog, det er interpreterende, hvilket betyder at oversættelsen, sker i medens kommandoen udføres. Det første sprog, der kunne sådan noget kaldte vi senere 'EXEC1'. Vi var meget glad for exec's. Vi sammenlignede dem med TSO's kommandofiler, og kunne så tydeligt se, at vi kunne mere, og langt bedre performance - den kørte altså hurtigere. VM *var* meget bedre!!!! Da EXEC2 kom, gik det op for os at EXEC1 performede *meget* dårligt, desuden havde EXEC2 en masse af de muligheder, vi manglede i EXEC1. Nu kunne vi med god samvittighed og begejstring fortælle vore kunder, at man med EXEC2 kunne lave *alt*. Så kom REXX!! Med REXX kan man lave meget mere, end man kan med EXEC2, og det er ikke rart at skulle fortælle at det man tidligere har sagt med stor overbevisning ikke passer, så der måtte en matematiker til at

berolige mig, han fortalte at uendelige størrelser ikke behøver at være lige store (jeg tror forresten han hed Jon!!!!).

Der var et lyst hoved der havde fundet ud af at det kunne være smart at sende et svar, hvis man modtog elektronisk post medens man ikke var til stede. Altså en slags telefonsvarer, noget med at meddelelsen var modtaget, men ikke kunne behandles foreløbigt. Den slags var let at kode i REXX, men der var visse ulemper. Det skete til tider at to virtuelle maskiner, de begge sendte den slags automatiske meddelelser begyndte at sende til hinanden, altså til stadighed. Joh, jeg er helt sikker på det kunne ske, jeg *er* blevet ringet op hjemme af en bekymret operatør, der syntes at to af mine maskiner producerede betænkeligt mange meddelelser.

Noget næsten tilsvarende skete, da vor linie til nettet blev flyttet fra Stockholm til Holland. Der var ske en fejl et sted under vejs således at filerne ét sted sendte filer til Danmark via Stockholm medens et andet sted undervejs sendte dem via Holland. Filerne blev altså sendt frem og tilbage - indtil fejlen blev rettet.

Naturligvis var der masser af uddannelse, kurser og seminars, men jeg husker især et kursus på NEC (Nordic Education Center) fordi det ikke havde med EDB at gøre. Jeg husker ikke kursets titel, det var måske "*Project Management 2*" men undertitlen var "*At arbejde i gruppe*". Vi fik lært en masse om, hvad der sker, når man arbejder i gruppe, og det blev demonstreret for os, at selv om vi var IBM'ere, reagerede vi nøjagtigt som instruktørerne forventede og altså som alle andre. Vi skulle f.eks. finde ud af hvilke ressourcer, der var vigtigst for en ekspedition på månen. Kravet var at gruppen kunne blive enige - der var *ikke* let. Joh, resultatet blev skam sammenlignet med 'rigtige' astronauters valg.

Vi fik til opgave at diskutere fordele og ulemper ved at dele en gruppes arbejdsopgaver i under-dele således at gruppens medlemmer hver fik ansvaret for en lille del. Der var én, der hævdede at det var en ulempe, fordi det for den enkelte kunne blive vigtigere at løse sin egen del, end at gruppens samlede mål blev nået. Det var der en pige, der protesterede voldsomt mod. Senere blev vi delt op på en anden måde - ved nogle borde. Vi fik hver en lille kuvert med brikker til et puslespil. Opgaven var at ved hvert bord skulle alle samle puslespil, sådan at alle havde et lille kvadrat. Tricket var naturligvis at det kun kunne lade sig gøre ved at vi byttede brikker indbyrdes - samtale var strengt forbudt. Mit bord blev hurtigt færdigt, fordi en norsk sælger fandt fidusen og demonstrativt kylede alle sine brikker ind midt på bordet. Det gjorde vi andre så også!!! Da vi var færdige, skulle vi så iagttage, hvad der skete ved de andre borde, og *der* så jeg førmtalte pige siddende ved et bord med sit færdige kvadrat, medens de andre rasende stirrede på hende, fordi de skulle bruge hendes brikker for at *de* kunne blive færdige. Så forstod hun endelig, hvad det drejede sig om.

Jon holdt flere VM-seminars - de fleste på NEC. Jeg husker han begyndte ét med at fortælle, hvad vi skulle høre om. Så sagde han at vi *hele* tiden var velkomne til at afbryde og stille spørgsmål. Han lovede at vi nok skulle få svar på alle de spørgsmål vi kunne finde på at stille. Jon var godt nok selvsikker, men svar på *alle* spørgsmål synes at være en stor mundfuld selv for ham. Joh, sagde han senere, han skulle nok svare på det hele, men det kunne da godt tænkes at hans svar ville blive "*Jeg ved det ikke*". Så var vi ligesom nede på jorden igen.

På et af disse seminars havde vi en masse hands-on - altså arbejde på maskinen. Og maskinen var *alt* for lille, det gik altså ualmindeligt langsomt.

Når VM ikke kører fullscreen - altså når den kører linie for linie, står der 'RUNNING' i nederste højre hjørne, undtagen når VM læser fra skærmen. Det var vi alle enige om var overdrevet, RUNNING altså. Jon benyttede så en sen aften (læs nat) til at lave en lille ændring, så 'RUNNING' blev til 'WALKING', 'CRAWLING' eller 'CREEPING' afhængig af hvor travlt maskinen havde. Bare for at være morsom, og sætte lidt kulør på tilværelsen.

Naturligvis virkede det ikke (ham der Myrphy igen)!!! Da Jon nu er en af de rigtig skrappe drenge, satte han sig til consollen og pillede lidt i maskinens hukommelse under VM's opstart. Det lykkedes. Han fik liv i systemet så det kunne køre normalt. Hvad Jon ikke vidste, var at maskinens tapestationer ikke virkede, så hvis hans anstrengelser under opstarten ikke havde givet det ønskede resultat, ville vi slet ikke have mulighed for at bruge maskinen mere, fordi vi ikke ville være i stand til at gendanne et kørende system.

Det ventede jeg naturligvis med at fortælle ham, til problemerne var løst. Man skal ikke stresser folk, der er lige ved at ligge ned.

Som jeg har fortalt, sker alting i VM i virtuelle maskiner. I en sådan maskine kan man køre alt, hvad der kan køre i en rigtig maskine. Man kan altså også køre VM!!! En eller anden skør fyr (det var vistnok mig), fik den strålende idé, at vi skulle prøve alt køre VM i mange lag. Altså VM under VM under VM o.s.v. Det gik rigtig fint. Jeg tror vi kom op på 5 niveauer. Men så gik det også rigtig langsomt - egentlig meget naturlig, når man tænker over, hvad der skal ske før en begivenhed i nederste lag, er blevet behandlet af alle de overliggende lag.

Til et andet seminar ankom jeg til NEC lidt sent på aftenen (altså dagen før). Så sent at jeg gik glip af det sene aftensmåltid. Jon, der boede i en funktionærlejlighed, inviterede mig så på aftensmad. Joh, han kunne sandelig lave mad. Ikke bare mad sådan noget tilfældigt rodet sammen, men sådan rigtigt!!! Medens vi spiste kom der et par andre og kiggede ind. Af en eller anden grund medbragte de alle en 'Gul Enke' (altså Champagne). Så inden aftenen (natten??) var gået, havde vi en formidabel champagne fest. Det er vist den eneste gang jeg har været i selskab med syv gule enker. Jeg ved ikke hvem der havde fået idéen med de der gule enker, jeg

er sikker på, jeg ikke havde hørt om det ellers havde jeg jo nok også haft en flaske, og måske havde jeg så også kendt årsagen til gildet.

Der var et af VM møderne i Stockholm, hvor vi fik anvendelse for en af VM's mere specielle muligheder. Samtidig med vort seminar var der et kursus for sælgerelever. Sådanne kurser er ret barske, disse unge mennesker skal naturligvis lære at livet i IBM godt kan medføre problemer. En af disse håbefulde havde en aften gjort et eller andet uartigt, så han skulle straffes. Deres instruktør kom og spurgte om vi havde et forslag. Og *det* havde vi. Vi ændrede en tabel i VM, så tasterne på konsollen fik en helt anden betydning således at et tastetryk medførte andre karakterer end dem, der stod på tasten. Jon gav så en pragtpræstation som en hysterisk kunde, der ikke kunne forstå noget som helst. Det kunne den stakkels sælgerelev bestemt heller ikke. Hans opgave var naturligvis at berolige den ophidsede kunde, og komme med en plausibel forklaring. Men da det var en straf, ville Jon naturligvis ikke lade sig berolige. Og en forklaring kunne der slet ikke blive tale om.

Min chef begyndte at synes at VM kørte vældig godt. Jeg fik at vide at det var på grund af mit store engagement og alt det der, men jeg fik altså at vide at jeg skulle have et nyt ansvarsområde, det er sikkert sådan en straf man får, når man har lavet et godt stykke arbejde. Det var ikke sjovt at sige farvel til VM - når man har arbejdet med VM er man tilbøjelig til at synes at alt andet er dårligere, hvad jeg bestemt også tror det er!!

Hvis jeg skal være *helt* ærlig, fik jeg vist først sagt VM farvel nogle år efter jeg blev pensioneret, og mit hjerte banker stadig varmt for dette dejlige produkt, som nu desværre er ved at være historie.

Der kom altså en ny VM-mand. Selvfølgelig skulle jeg sætte ham ind i arbejdet, og da jeg stadig var i afdelingen, skulle jeg naturligvis stå ham bi, med råd og dåd. Den første dag, begyndte jeg at vise ham lidt om, hvordan VM var 'skruet' sammen. Vi kiggede på de privilegerede maskiner, og på hvordan brugerne blev defineret. Det lykkedes mig ved en helt utrolig klodsethed at få slettet VM's allervigtigste data, nemlig den fil, hvor brugerne og brugernes privilegier er defineret. Så fik den nye mand et udmærket eksempel på, hvordan man redder stumperne. I løbet af et par timer, fik jeg faktisk lavet et nyt 'directory', så VM også fremover kunne køre. Naturligvis har jeg siden haft megen kontakt med denne 'nye' mand (der jo ikke mere er særlig ny), og han har da fortalt mig at mit lille uheld faktisk havde været utroligt lærerigt for ham. Mine anstrengelser var altså ikke helt spildte, men jeg er stadig meget flov over at jeg i den grad dummede mig.

Under MVS var der kommet et interaktivt produkt, der hed VSPC (Virtual Storage Personal Computing). Det er med overlæg, det er nævnt i datid, for det blev aldrig

nogen succes (det er vistnok *ikke* udelukkende min skyld, men jeg tror nok jeg gjorde hvad jeg kunne), så det er forlængst gået i glemmebogen, efter min mening særdeles velfortjent.

System-vedligeholdelse under MVS forgik (og foregår stadig) ved hjælp af TSO. I de dage kørte TSO meget langsomt, især når man sammenlignede med VM/CMS (jeg kan endnu en gang konstatere at VM er et fremragende system), så jeg gik i gang med at skrive nogle EXEC-procedurer, der gjorde det muligt og bekvemt for mig at, at lave VSPC vedligeholdelsen fra CMS. Det gik fint. Jeg kunne stadig bruge mit kære VM system, og alligevel passe mit MVS relaterede arbejde. Jeg fortalte en kollega om disse rutiner og da han mente at der sikkert var andre, der kunne få glæde af dem, placerede jeg dem på en offentlig disk - altså én alle havde adgang til. Telefonen ringede næsten øjeblikkeligt. Det var én, der havde fundet en fejl, og når folk ringer og beklager sig over fejl i programmer, er det tegn på at de bruger disse programmer. VSPC blev dog aldrig noget jeg kunne lide, jeg kan vænne mig til meget, men altså ikke til VSPC. Jeg agiterede kraftigt for at man skulle smide dette - efter min mening - forfærdelige og underlødige system ud, og flytte hele butikken over på CMS. Det var nok ikke på grund af mig, for min daværende chef var vildt uenig med mig (jeg har godt nok aldrig anset ham for at være en særlig god chef), men det var helt nøjagtigt, hvad der senere skete. Jeg tror (ved) det har været en *meget* stor fordel for IBM at man skiftede sin interaktive interne databehandling til VM, ikke mindst takket være VNET.

Computer-centeret blev slået sammen med DP's testcenter, og det gav naturligvis nogle problemer. Når to afdelinger bliver lagt sammen, er der altid nogle stillinger, der er dobbelt besat, og hvem skal så . . .

Vor afdelingschef fik imidlertid en strålende idé. Han inviterede os alle til Rungstedgård et par dage. Meningen var naturligvis at vi skulle lære hinanden bedre at kende ved forskellige gruppearbejder. Én af opgaverne var at vi i grupper skulle definere vore arbejdsproblemer. Det kom der rigtig mange morsomme diskussioner ud af. Der var blandt andet en del, der havde deres leder som største problem - måske var det ikke særligt morsomt for disse ledere. Der var dog også mere praktiske opgaver. Jeg var f.eks. med i en gruppe (et udvalg??), der skulle komme med forslag til indretning af lokaler i forbindelse med en 'drop-in' service, altså noget med at man kunne komme og aflevere sine jobs ved en skranke og forhåbentlig hente resultaterne senere. Det var vældig spændende. Et af vore problemer var vinduer. Der er i Danmark regler om at man skal have vinduer i lokaler, hvor mennesker har deres arbejdsplads, og jo større lokale des større vinduer. Vi fik dog lavet en god plan, der blev nedfældet i en vældig fin rapport (med rødt omslag). Det blev aldrig til noget, men det var meget spændende at være med til måske fordi det var noget der *næsten* ikke havde med jobbet at gøre - altså en behagelig afveksling. At det ikke blev til noget gjorde mindre, jeg har gennem årene lavet mange fornøjelige og ind imellem avancerede ting der aldrig er blevet

til noget, det vigtigste er at man har det sjovt medens man laver det, og naturligvis at man har en fornemmelse af at have gjort et godt stykke arbejde.

En skønne dag fik vi også en ny afdelingschef, der havde sine egne ledere, så vi fik også en ny afdelingsleder. Hans og min opfattelse af databehandling var mildt sagt forskellige. Han var en flink fyr, men efter min mening en elendig leder - det har jeg vist nævnt tidligere. Jeg må erkende at han gjorde rigtig meget for at få samarbejdet til at fungere, måske fordi han havde hørt at jeg ikke var udpræget dum til mit arbejde. Han sendte mig på konvention, altså en tur, der svarede til turen til Lissabon. Nu var vi imidlertid blevet opslugt af en administrativ afdeling, så dette rejsemål var knapt så eksotisk, det gik til Rönneby i Syd-Skåne.

Der var en masse mennesker, jeg ikke kendte og der skete egentlig ikke ret meget jeg fandt interessant. En aften stod middagen på pattegris på spid (Sådan én er faktisk ret stor). Vi sad udendørs, og det så egentlig ret lækkert ud - med rødvin-flasker på bordet. Grisen smagte dejligt, der var masser af kød, flutes og salat, men da vi fandt ud af at Kirsten W., havde fødselsdag, var der ikke så meget rødvin tilbage, at vi kunne skåle med hende, og det synes jeg egentlig ikke var rigtig op til IBM's standard.

En pudsig episode indtraf under en middag. Vor personalechef var naturligvis med, og han havde åbenbart fået lidt rigeligt at drikke, så efter middagen, sad han i et hjørne, og så *meget* træt ud. En af operatørerne spurgte mig hvem den der fylderist var. Jeg svarede at det var for dårligt at han ikke kendte sin personalechef, men fik at vide at det kunne da være ligegyldigt, personalechefen kendte nok heller ikke ham!!! Jeg gik så hen og ruskede lidt i chefen, og fik ham faktisk gjort interesseret, "Hva' er der???" Joh, sagde jeg, *ham der* tror ikke du kender ham. "Hvem Johnny???" lød det fra chefen. Naturligvis havde vor personalechef interesseret sig for, hvem der skulle med, så han 'kendte' os alle sammen.

Sammen med turen fik jeg et Cross pencil sæt i gulddublé. Det er vældig fint (og dyrt), så jeg nænner ikke at gå med det - det har ligget i en skuffe næsten lige siden jeg fik det.

Der var en mere positiv oplevelse. Afdelingen med ledsagere, blev inviteret på 'Time Manager' kursus. Det foregik på Hotel Marienlyst i Helsingør, et dejligt sted med gode værelser og god mad, og et Time Manager kursus er ret underholdende. Både Kirsten og jeg havde et par dejlige dage. Pudsigt nok så jeg et eksempel på Time Manager undervisningen i fjernsynet, som det første, da vi kom hjem.

Jeg kunne vældig godt lide Time Manager, men jeg skiftede dog hurtigt til TimeSystem, da jeg synes dets A5 format er mere praktisk, f.eks. kan man let fotokopiere en A4 side til halv størrelse og indsætte den i TimeSystem, noget jeg har brugt rigtig meget.

Det er ikke rart at have et dårligt forhold til sin chef, det er heller ikke rart for chefen, så da NEUCC på Danmarks Tekniske Højskole (nu Danmarks Tekniske Universitet) besluttede at de ville supplere, deres MVS system, der var ved at blive overbelastet, med et lille VM system, fandt man ud af jeg passende kunne assistere den pågældende salgsafdeling, der ikke selv havde VM ekspertise. Det var en strålende idé. Jeg havde det *pragfuldt* sammen med de flinke DTU'ere, og min chef var fri for mig en del af tiden. Det var lidt af et kulturchok. Jeg skulle nu for første gang siden 1620-tiden til at have med kunder at gøre. Jeg tog slips på til det første møde, for at være på den sikre side. Der var en af folkene fra DTU, der sad kun iført et par små bade shorts (det *var* sommer og hedebløge), så det var sidste gang jeg havde slips på der. Arbejdet på DTU, var meget lig, det jeg gjorde på IBM, blot var der *her* folk til det grove, min opgave var at komme med gode råd og løsninger. Det voldte ikke mange problemer, det var en særdeles dygtig kunde, og *jeg* havde et glimrende forhold til teknisk afdeling. Den tekniker, jeg i sin tid hjalp med et meget besværligt HASP/APL problem, var i mellemtiden blevet VM tekniker!!

Når jeg omtalte DTU's nye VM system som et lille system, var det dels fordi DTU's MVS maskine var en virkelig sværvægter, men også fordi den nye maskine var en helt ny teknologi, så den rent faktisk ikke fyldte ret meget. En morgen, blev jeg slæbt ned i maskinstuen af en grinende projektleder. Han spurgte om jeg kunne finde den nye maskine. Nu havde jeg på det tidspunkt set en hel del maskinstuer, så *det* troede jeg da nok jeg kunne. Det viste sig dog at den nye maskine, var en lille grå boks, som jeg helt bestemte troede var et airconditionlæg eller sådan noget. Den synede af *meget* lidt ved siden af de store 'gamle' maskiner.

Jeg var glad for at arbejde for DTU, og DTU var glad for min hjælp, så da projektet var afsluttet, fik jeg en vældig pæn skrivelse fra salgsafdelingen, hvor jeg nærmest blev rost til skyerne. Der blev naturligvis sendt en kopi til min chef. Han nævnte den aldrig - det det syntes han ikke der var grund til, jeg havde jo fået den selv. Sådan skal en god chef - efter min mening - *ikke* opføre sig.

Når man ikke har det godt med sin chef, og man lige har haft succes med en kunde i felten, er det naturligt begyndte at lede efter nye muligheder netop i felten - altså i en salgsafdeling. Det var på de tider, hvor IBM gerne ville have de små kunder, der kørte VSE (VSE er den moderne udgave af det gode gamle DOS system), til at køre på VM, med VSE som gæste operativsystem. Fordelen ved det var at man med VM kunne udnytte større maskiner end med VSE, og på den måde få adgang til større kapacitet. Dette indebar naturligvis at salgsafdelingerne havde brug for VM kundskab - altså netop sådan én som mig, så jeg prøvede at komme til en salgsafdeling, som VM specialist.

Nu er det at flytte en mand ikke altid noget, der går let. På trods af det dårlige forhold mig og chefen imellem, blev jeg stadig anset for at være en *god* mand, så

man ville gerne have noget til gengæld. Jeg ved ikke om man fik det, men der gik et halvt år, fra det var besluttet at jeg skulle flyttes, til det rent faktisk skete. Da alle vidste, jeg var på vej væk, fik jeg ingen rigtige opgaver - jeg havde altså ikke rigtig noget at lave. Hvad gør man så??? Da jeg havde adgang til en computer med VM, gik jeg i gang med at forfine mine exec-procedurer, og skrive nogle nye - dels for at få tiden til at gå, og dels for at dygtiggøre mig selv. Jeg gik også i gang med at undersøge forskellige tools-programmer, og fik en hel del korrespondance med forfatterne, enten fordi jeg fandt fejl eller fordi jeg havde forslag til væsentlige forbedringer.

Endelig fandt Systems Marketing ud af at de godt kunne bruge en mand med kendskab til VM. Det var helt fint, bortset fra at Systems Marketing havde til huse i midten af byen (Vester Søgade), så jeg, i stedet for en daglig tur fra Kokkedal til Lundtofte, fik en daglig tur fra Kokkedal til Vester Søgade. Det var dobbelt så langt og ydermere altså til midten af byen. Heldigvis kører der busser fra Kokkedal til midten af København, så jeg begyndte at interessere mig for offentlig transport.

Hvor var det dog dejligt at komme væk fra computercenteret (hvor jeg havde haft rigtig mange gode år) og det var dejligt at få noget lave igen. Jeg tror ikke ledig-gang er roden til noget som helst dårligt, men hvis man *skal* være på kontoret, er det altså bedst at have et formål med at være der.

Alt i alt syntes jeg dog at jeg endnu en gang havde været heldig!!! Jeg skulle igen til at arbejde med og for mit kære VM system, og fremover skulle det altså gå ud over *rigtige* kunder. Den bedste måde at gavne et system som VM, er altså at få andre til at interessere sig for det, så det var det jeg energisk prøvede at gøre. Jeg tror i hvert fald ikke nogle af mine kunder nogensinde har været i tvivl om at jeg selv var utrolig glad for VM.

This page intentionally left blank.

VM i marketing

Så var jeg havnet i en salgsafdeling. Da salg er det IBM lever af, er salgsafdelingerne naturligvis nogle af de vigtigste afdelinger i firmaet, så det var en ganske rar fornemmelse. Jeg lærte hurtigt at kunderne altid har ret - hør, det var da vist noget sludder. Set fra mit (tekniske) synspunkt skulle kunderne normalt have hjælp bl.a. når de havde gjort noget galt/forkert, og når man har gjort noget galt/forkert, har man jo netop *ikke* ret, men hjælp fik de.

Desværre var 'min' nye afdeling placeret i Vester Søgade altså midt i København, så jeg fik pludselig dobbelt så langt til arbejde - og *så* inde midt i byen. I starten lod jeg bilen stå hjemme, jeg fandt ud af at det slet ikke er så slemt at køre med bus, og når det regnede eller var snestorm, var de 10 - 15 minutters gang til busstoppestedet ikke særlig rare, men vejret var da heldigvis ikke altid sådan. Det var ikke parkeringen inde i byen, der gav problemer, der var en stor underjordisk parkeringsplads under kontoret, og da IBM gerne så vi hurtigt kunne komme ud til kunderne, betalte de gladelig parkeringsafgiften.

Kontoret var ikke specielt behageligt, det var faktisk ret ubehageligt. Om sommeren kunne vi nå op på temperaturer, der var en hel del over 30°C. Jeg sad et stykke tid i et sydvendt kontor med meget store vinduer, og den eneste afskærmning for solen var et sort gardin, der mest af alt mindede om krigens mørklægningsgardiner, altså et tæt sort ugenomsigtigt stykke stof. I sandhed en sær fornemmelse at sidde på et kontor, kun oplyst af kunstigt lys, med kulsorte gardiner for vinduerne, når det var det mest vidunderlige sommervejr udenfor. Det er utroligt et så forholdsvis nyt kontorhus er bygget så det i den grad er uegnet til sit formål.

Jeg fik omgående et - efter min mening - kolossalt stort problem. Jeg fik ingen skærmterminal!!! Der var slet ikke skærme til alle i afdelingen, der var faktisk meget få skærme, men afdelingen lå jo også langt fra maskinerne, der jo stadig var placerede i Lundtofte, og jeg var jo ny og derfor ikke specielt privilegeret. Endelig er jeg slet ikke sikker på der i Vester Søgade kontorene var skærme på folks skriveborde dengang. Nu havde jeg boltret mig som systemprogrammør i ganske mange år, naturligvis med egen skærm, og så pludselig *ingen terminal*. Det snakkede jeg *meget* med min chef om, rigtig meget, jeg var vel så vedholdende som Cato, der i sin tid ved enhver lejlighed snakkede om at ødelægge Kartago. Så en dag kom chefen stolt ind på kontoret for at invitere mig ud på gangen. Der stod en enorm stor papkasse, som viste sig at indeholde - ikke bare en skærm - men en splinter ny farveskærm, den model, der på den tid var '*state of the art*'. Den var endnu bedre end min skærm i Lundtofte, der 'kun' var en monokrom. Jeg husker stadig hvordan jeg pakkede den ud og placerede den på mit skrivebord, der omgående var ved at bryde sammen under vægten, sådan en skærm var nu også tung, ikke bare tung, men rigtig tung, og det ene bordben var simpelthen ikke skruet ordentligt på. Heldigvis var mine reflekser i orden, så jeg undgik en katastrofe, og problemet kunne let løses med min trofaste schweizerkniv. Skærmen tjente mig tro i mange år.

Jeg mistede den desværre rigtig mange år senere under en af de mange flytninger fordi jeg havde glemt at sætte en adresseseddel på skærmen, jeg *havde* sat én på terminalbordet, men det var altså ikke tilstrækkeligt. Snart efter fik jeg imidlertid tilbudt en lille PC i stedet, så det var et tab, der var til at overse - og dog, på *det* tidspunkt var denne skærmtype så gammel at de fleste ikke mere virkede særlig godt, men min var altså i tip top stand, og den var sandsynligvis en af sidste, der kunne køre nogenlunde fornuftig, den erstatnings skærm jeg fik, var i hvert fald ikke særlig rar at arbejde med.

Den fine nye skærm havde ikke rigtig grafik, men man kunne ændre dens tegnsæt. Man kunne f.eks. lave meget specielle tegnsæt, hvor tegnene var små dele af et billede og på den måde opnå noget, der mindede om grafik. Naturligvis var der programmer, der klarede sådan noget, så det var altså muligt at vise noget, der mindede om fotografier på sådan en skærm. Billederne blev dog aldrig særligt store, det var jo tegnsæt, der opbyggede billedet af disse tegn og der var vistnok kun plads til et par tegnsæt af gangen. Jeg husker jeg havde fundet et 'fotografi' af en nydelig ung dame i en tynd hvid badedragt. Da min afdelingschef tilfældigt kom forbi en gang jeg havde hende på skærmen, kaldte jeg på ham. Han kiggede længe på den nydelige pige - og derefter på tastaturet. Så spurgte han: "Hvor er den knap, der gør hende våd?"

Nu er det at fungere i en salgsafdeling noget helt andet end det, jeg havde beskæftiget mig med før, det er vist ikke helt rigtigt, i min 1620-tid arbejdede jeg jo også med kunder, men nok på et helt andet niveau. Min nye afdelingsleder fik desværre den ulyksalige idé at jeg skulle på grunduddannelse!! Det var naturligvis fordi jeg skulle lære at omgås kunder på den 'rigtige' måde. Jeg var 48 år og havde 20 års erfaring i IBM. Jeg kan garantere at det ikke er den bedste baggrund for en grunduddannelse. Jeg tror nok jeg fik lært lidt, men ind imellem kneb det med at tage det alvorligt. Især havde jeg problemer med hjernevasken - det burde jeg nok ikke kalde det, men jeg følte at det var *det* man prøvede. Et af formålene med grunduddannelsen var naturligvis at alle de kommende 'unge løver' skulle tilpasses IBM eller 'Big Blue', de skulle altså 'males' blå. Et andet formål var at vænne disse nye IBM'ere til den stress, der så let kommer i kundesituationer og måske især i forbindelse med arbejde med computere. Selvfølgelig havde jeg ofte været udsat for lagt mere stressende ting end man kan opfinde på et kursus, så heller ikke det kunne imponere mig.

Jeg kom næsten helskindet gennem et par kurser, og fortalte naturligvis min nye chef, hvad jeg mente om det. Han var heldigvis en flink og fornuftig fyr, så han lod nåde gå for ret, så jeg slap for resten af turene til Stockholm, hvor grunduddannelsen foregik. En af fordelene ved at være på kurser i Stockholm var naturligvis at det gav mulighed for en flaske skattefri sprit, men også den slags kan købes for dyrt og jeg synes selv jeg kom rimelig godt gennem resten af min IBM-tid uden at have afsluttet denne grunduddannelse.

I starten syntes jeg nok min nye chef holdt lovlig meget øje med, hvad jeg foretog mig. Jeg var slet ikke vant til at have én til at kigge mig overskulderen hele tiden, og han kendte mig da fra tidligere, jeg havde (naturligvis) ofte hjulpet ham med VM-problemer vistnok bl.a. i forbindelse med et kursus han holdt, og jeg var jo som specialist vant til at arbejde med ting jeg vidste langt mere om, end min chef. Men heldigvis fik han ret hurtig tillid til at jeg brugte min tid rimeligt fornuftigt, og så fik jeg stort set lov til at gøre, hvad jeg havde lyst til.

Systems Marketing, som jeg var havnet i, betjente små kunder, der benyttede små men 'rigtige' EDB maskiner. Jo mindre kunderne er, jo mere hjælp har de brug for, naturligvis fordi de ikke selv har råd til specialister på alle områder. Disse små kunder kørte (næsten) alle VSE, altså *det*, den gode gamle DOS havde udviklet sig til. System Marketing havde en VM-mand, da jeg kom, så planen var vistnok, at jeg både skulle hjælpe VSE og VM kunder med problemer. En håbløs idé. Godt nok havde jeg arbejdet rigtig meget med DOS i mine yngre dage, men der var sandelig løbet meget vand i stranden siden, både set fra mit og DOS's synspunkt. Heldigvis erkendte man hurtigt at jeg var mere produktiv på VM og da den anden VM-mand vidste mindre om VM end jeg,, blev jeg *VM specialisten i Systems Marketing*. Han havde jo ikke været systemprogrammør og samtidig var han meget mere interesseret i teknisk databehandling altså CAD/CAM og sådan noget, så vi var vistnok begge godt tilfreds med den arbejdsfordeling, og jeg tror bestemt det var en stor fordel både for ham, for mig og for SM at arbejdet blev fordelt på denne måde.

Jeg kom lige tilpas til SM. Man havde solgt en lille VM maskine til et analyse institut. Dette institut skulle udskrive nogle tabeller på en helt ny type printer - en 6670. En 6670 var en IBM kopimaskine, der kunne kommunikere og således anvendes som en laserprinter. Sammenlignet med, hvad vi havde haft af printere inden da, skrev den vældig pænt, men den var nu også ret stor. VM bruger, som tidligere nævnt, programmet SCRIPT, når man skal skive pænt, og SCRIPT havde jeg en del erfaring med, også fordi jeg altid har syntes at tekstbehandling er spændende - men det har *aldrig* været mit arbejdsområde. 6670 var en tekstbehandlings-maskine, altså ingen normal printer - set fra VM's synspunkt, så den krævede særlige kundskaber. Nu er IBM et firma, hvor man altid kan finde nogen, der ved noget om det, man har brug for. 6670 specialisten jeg skulle arbejde sammen med hed Dorthe og kom fra Ålborg. Hun blev indlogeret på et hotel, og vi blev anbragt i et fjernt kontor, for sammen at løse denne opgave i fred og ro.

Det blev en besynderlig erfaring. Vi kunne vældig godt sammen - det er i øvrigt meget få gange, jeg har mødt en IBM'er som jeg *ikke* kunne arbejde sammen med - men vi fandt ud af at vi *slet* ikke talte samme sprog. Joh, selvfølgelig kunne vi begge dansk og naturligvis en hel del engelsk, men tekstbehandlere kalder ting det samme, som vi andre, men de mener noget *helt* andet. Det gav anledning til en hel

del hede diskussioner. Vi havde begge temperament, så vi blev irriteret, når den anden ikke kunne forstå meningen. Kommunikation mellem mennesker er virkelig noget meget vanskeligt, men det er det jo også mellem maskiner. Opgaven blev dog efterhånden løst, kunden blev særdeles tilfreds, så alle var glade. Så glade, at jeg for første gang stiftede bekendtskab med den udmærkede opfindelse man kalder en 'Middagsaward'. Det går i al enkelthed ud på at man kan invitere sin kone (eller mand) ud for at spise en bedre middag - på firmaets regning. Det er vældig rart, når man har lidt dårlig samvittighed over ofte at have forsømt fruen på grund af alt for meget overarbejde. Det er selvfølgelig også rart ikke at behøve at spekulere alt for meget over regningens størrelse. Det blev Vig Kro, der fik lov at forkæle os, en gang vi var i sommerhuset, jeg husker stadig at min søn Claus for første gang meget gerne ville prøve at spise snele!!!

På den tid begyndte man at tale om PC'ere og dermed også om computer-vira. Det grinede vi meget af, sådan noget kunne man jo slet ikke på VM, der jo er et fuldstændigt sikkert system!!!!

Nej, rigtige vira havde kun ringe/ingen chancer overfor et system som VM, men computercenteret blev alligevel ramt af noget der lignede en virus. Der kom en dag - med den elektroniske post - en lille exec-procedure, ved navn XMAS (altså en slags julekort). Sådan noget havde vi set så tit, det var sædvanligvis noget med at tegne et julekort eller sende en anden fornøjelig julehilsen. Jeg fik den selv, og kørte den, vi troede jo alle at folk i vor VM-verden var flinke og rare. Pludselig begyndte der at ske ting og sager, så jeg fik omgående standset den. Sådan gik det ikke for alle. Denne ondsksfulde exec, gik ned i brugerens adresse-fil (sådan én findes på alle VM maskiner), hvorefter den sendte sig selv til alle dem, den kunne finde i denne fil. På den måde kunne den sprede sig utrolig hurtigt - også over landegrænser. Den gang var der en øvre grænse for hvor mange elektroniske breve (filer) der kunne opbevares i et VM system - og der var naturligvis også begrænset diskplads, så i løbet af næsten igen tid gav VM op, og først efter indgreb fra operationen, kom systemet i gang igen. Indgrebet bestod bl.a. af at man foretog et hurtigt eftersyn af alle brugernes diske og slettede *alt*, hvad der lignede XMAS, og naturligvis lavede man derefter et filter, der frasorterede, hvad der fremover kom af den slags.

Selvfulgelig blev der sat en undersøgelse i gang - sådan noget kan man naturligvis ikke lide. Det viste sig det var Country SE Manager (altså chefen for alle system-konsulenterne), der havde spredt den i Danmark. Den stammede i øvrigt fra et amerikansk universitet, og var kommet fra universiteternes computer-net til IBM's netværk gennem en gateway altså en forbindelse mellem de to netværk.

Jeg fik installeret en masse VM-systemer, men det var ikke dem alle der gik helt efter bogen. Jeg husker jeg skulle til Fredericia en weekend for at installere et system. Jeg fyldte nogle tapes og programmer i min bil, og begav mig af sted.

Opgaven var helt enkelt, jeg skulle læse nogle back-up tapes ned på et par diske, og tilpasse systemet til kundens maskine. Det at lægge en backup tape ned på en disk, når man ikke har noget system, gøres med et program, der findes på en stak hulkort. Alt var rutine-arbejde - noget jeg havde gjort masser af gange før - men denne aften - måske fordi det var lørdag - kom der læsefejl på tape. Hvad gør man i Fredericia en lørdag aften, med en tape man ikke kan læse?? Jeg havde bestemt ikke lyst til at spolere endnu en weekend med samme projekt. Først rensede vi tapestationen, sådan en bliver naturligvis også snavset fra tid tilanden. Så prøvede vi en gang til med samme nedslående resultat. Heldigvis fandt jeg ud af at jeg havde fået lagt tape nok ned på disken, til at den indeholdt et VMsystem der kunne startes. Det gjorde jeg så. En VM'er har det altid bedst, når han har adgang til et rigtigt VM-system. Det viste sig at VM programmet til at læse tapes var rigtig meget bedre end hulkortprogrammet, således at jeg nu uden problemer kunne læse *hele* tape. Det er fordi et lille hulkortprograms fejlbetringelses rutiner er meget mere primitive end de rutiner, der ligger på et 'rigtigt' system. Lørdagen og kundens VM var reddet, og jeg og VM kunne notere endnu en sejr. Det er som om man har det endnu bedre, når det var *lige* ved at gå galt, og man så klarer den, joh jeg var vistnok temmelig glad for mig selv.

Flytninger var noget vi blev udsat ustandselig, naturligvis altid i forbindelse med årsskifte, men i løbet af året var det slet ikke ualmindeligt at man pludselig skulle have nyt kontor. Jeg husker dog én gang der var *lidt* usædvanlig. Da jeg kom tilbage fra ferie og skulle indtage min plads bag skrivebordet, var kontoret væk!! Der myldrede med håndværkere, gamle vægge var revet ned, og nye var ved at blive bygget. Jeg måtte lede længe, ja faktisk skulle jeg en etage op, for at finde mit skrivebord.

En dag skulle jeg installere et VM system i Randers. Det foregik i et helt nyt hus - nede i kælderen. Jeg husker stadig at EDB-chefen havde et vældig stort men ikke uvæsentligt problem, kunne han skaffe kaffe til os alle??? Det kunne han heldigvis. Da jeg begyndte min installation, fik jeg imidlertid et meget større problem. Jeg læste mine tapes ned på disken, og skulle så starte VM, for at tilpasse den, til netop denne maskine, altså rent rutinearbejde. Jeg startede VM, men der skete *intet* på den skærm, jeg regnede med at bruge til kommunikation med VM. Mystisk, dette var noget jeg havde gjort masser af gange - helt uden problemer. Hvad gør man?? Det vigtigste er *ikke* at viste kunden at man er lige ved at bryde sammen i panik. Sammenbrud er måske lige i overkanten, men når man kommer som specialist, skal tingene jo helst lykkes og det smertefrit. Når man ikke har bedre idéer, kan man jo altid prøve igen. Ikke fordi det plejer at hjælpe, men det giver én tid til at tænke sig om en ekstra gang. Netop som jeg kom i tanke om at VM bruger den tændte skærm, med den laveste adresse, som konsol - altså til kommunikation med operatøren, og det derfor kunne tænkes at der var en skærm på et tilfældigt kontor, der var tændt,

kom der en fyr styrtende ind i lokalet og fortalte at der stod noget besynderligt på skærmen på hans kontor. Det 'besynderlige' var netop VM's logo, så det var *den* skærm, VM ville bruge som konsol i stedet for den, der stod ved maskinen, lidt kedeligt, men VM har vel ingen mulighed for at mærke hvilken skærm, der er placeret ved maskinen. Problemet var altså det enkle at jeg burde styrte rundt i hele huset og slukke samtlige skærme for at være *helt* sikker.

Endnu en gang blev det til en vellykket VM-installation, og der er nok ikke andre end mig, der lagde mærke til det 'lille' problem, der var meget tæt på at give mig et større nervesammenbrud.

Jeg fik efterhånden en hel del erfaring i at installere VM på toppen af VSE systemer. Jeg havde det vældig morsomt, og jeg fornemmede at kunderne syntes om mig, og det er bestemt ikke nogen dårlig fornemmelse, når man nu er i en salgsafdeling. I VM-verdenen er det sådan at Poughkeepsie er stedet, hvor alting foregår, så naturligvis drømte jeg om at se Pookie (Poughkeepsie er ikke særlig let at stave til, så mellem venner kaldes det ofte Pookie). Jon var kommet hjem fra Greenford, og han lavede en aftale med en af vennerne fra Stockholm, der var fast udstationeret i Pookie - altså på assignment. De syntes jeg havde gjort det så godt i VM verdenen, at jeg fortjente en tur til staterne, så de blev enige om at der skulle laves et residency (det er en speciel opgave, der skal løses af et hold af IBM specialister, der bliver samlet på stedet) specielt for at give mig lejlighed til at komme 'over there'.

Jeg vidste ikke at dette residency var lavet for *min* skyld, men da jeg så, hvad det drejede sig om, kunne jeg naturligvis godt se, at det *lige* var mig. Jeg kontaktede altså chefen og forelagde ham min mening. Han var ikke uenig, så han gik til sin chef, og vendte så tilbage og fortalte at jeg havde *så* travlt og var *så* uundværlig (???), at hvis *han* sendte mig til Poughkeepsie, ville han selv blive sendt til Sibirien eller et endnu værre sted. Nå, man kan ikke have succes hver gang, og der var da også andet end U. S. A. o.s.v. - alt i alt lettere skuffet, måske endda lidt mere end lettere, men i IBM hjælper det ikke at blive sur, og livet skal jo gå videre.

Et par dage senere så jeg tilfældig Jon's chef vandre meget beslutsomt ind til min afdelingschef, altså min chefs chef. Hvorfor beslutsomt, jo jeg kendte ham vældig godt, jeg havde hjulpet ham med adskillige VM-problemer, så vi stod på særdeles god fod, og han havde slet ikke tid til at hilse!! Hans tilstedeværelse kunne selvfølgelig betyde hvad som helst, men der gik ikke så lang tid før min afdelingsleder kaldte mig ind på sit kontor og begyndte at forklare mig at jobbet i Pookie lige netop var mig, at det var meget vigtigt at netop de rigtige IBM'ere deltog i den slags arbejde, for at det skulle blive så vellykket som muligt, så han kunne slet ikke se hvordan jeg kunne undgå at deltage En værre gang sludder, men han ville vel ikke tabe ansigt, men jeg fik altså lov at drage til staterne, og det var jeg bestemt ikke ked af.

En venlig kollega, der havde tilbragt et par år i Pookie, fortalte mig, hvordan jeg skulle finde mit hotel, hvordan man skulle gebærde sig henne i USA og, hvad der var *meget* vigtigt, at biler i USA har en lås, man skal aktivere, før man kan fjerne tændingsnøglen. Jeg fik også at vide at el stik i USA, er *helt* anderledes, end de hjemlige, og at man godt må dreje til højre ved rødt lys, hvis der altså er plads i trafikken og ingen skilte, der forbyder det.. Det var alt i alt meget vigtige/praktiske oplysninger, og det meste fik vi da også fortalt af den lokale sekretær allerede den første dag. Men det der med bilen, hotellet og el-stikkene var nu rart at vide på forhånd.

Amerika *er* et stort land, og jeg var *meget* spændt. Jeg fik mig et visum. Det er ikke svært, når man er ansat i et anerkendt amerikansk firma, men det var altså nødvendigt den gang. Jeg pakkede min kuffert og drog af sted. Sådan en jumbojet er ret stor. Det var næsten som at komme ind i en biograf. En venlig dame så på billetten, og viste så, hvilken række man skulle gå ned af. Jeg tror stadig ikke på at sådan en stor maskine, kan flyve. Det *kan* simpelthen ikke lade sig gøre. Jeg tror de sætter den på en blokvogn og kører den til bestemmelsesstedet, naturligvis med en film uden for vinduerne, så det ser rigtigt ud!!! Jeg kom til Kennedy Airport (JFK), og blev modtaget af en venlig SAS funktionær, der endog kunne lidt dansk. Han ville se min billet, for at hjælpe mig videre. Jeg skulle flyve med *Command Airlines*, og *det* havde han aldrig hørt om, så jeg fik at vide at der i ankomsthallen var en skranke med en dame, der vidste *alt om flyselskaber*. Hun så lidt skræmt ud, da jeg sagde *Command Airlines*, men hun kiggede i en meget stor bog, og fortalte at *Command Airlines* afgik fra TWA's terminal. Jeg gik udenfor, og så at TWA havde to terminaler, en international og en domestic - altså indenrigs. Indenrigs var naturligvis længst væk, men en tur til Pookie fra New York, altså ca. 150-km, kunne vel næppe være udenrigs, så jeg traskede af sted. Der var en *meget* flot negerpige i TWA's indenrigsterminal, men hun havde heller *aldrig* hørt om *Command Airlines*. Også hun så i den der store bog, og henviste mig til Lingus Airlines (det irske). Der var de helt sikre på et det var TWA udenrigs, så endnu en travetur. TWA udenrigs er en meget flot betonkonstruktion, og jeg henvendte mig til en uniformeret fyr, der så enormt selvhøjtidelig ud - amerikanere har et eller andet med uniformer. Men *Command Airlines* havde han aldrig hørt om. Heldigvis så jeg et skilt med *Command Airlines* lige bag ham, så vidste jeg, jeg var på rette vej. Når man kommer fra lille Danmark, er det ikke let at forstå, at et indenrigsflyselskab kan være *så* ukendt, i en lufthavnsterminal, som det beflyver flere gange om dagen, men der *er* altså store forhold.

I Pookie kaldte de flyveren fra New York til Pokeepsie, for Kamikaze. Det var en lille propel-maskine, der mest af alt lignede en udrangeret jernbanevogn med et bræt liggende på tværs ovenpå som vinger. Jeg steg om bord med en frakke over armen og en mappe i den anden hånd. Da jeg fandt min plads, hjalp pigen ved

vinduespladsen mig med at fjerne sikkerhedsselerne, der lå smukt over kors på mit sæde, og så faldt vi naturligvis i snak. Jeg er ingen ørn til engelsk, så hun spurgte naturligt nok, hvor jeg kom fra. Hun var jøde, og havde interesseret sig meget for, hvordan jøder blev behandlet i Europa under krigen. Heldigvis for mig, havde hun et vældigt godt indtryk af danskere - netop fordi vi *ikke* ville acceptere at jøderne blev udryddede, og arbejdede flittigt på at få dem sendt til Sverige. Da hun spurgte hvad jeg skulle så langt væk hjemme fra, og jeg svarede: "Arbejde", sagde hun: "Så må du være fra IBM, det er *jeg* også". Jeg syntes naturligvis det var et pudsigt tilfælde, men så fortalte hun at besætningen nok var de *eneste* i denne flyver, der *ikke* var ansat i IBM. Det var dengang det rigtigt gik op for mig at jeg arbejdede i et vældigt stort multinationalt firma.

Da vi ankom til Pookie, stillede jeg mig som sædvanlig op for at vente på min kuffert. Den kom ikke. Det var første og eneste gang jeg har været ude at rejse uden min kuffert nåede frem sammen med mig. Ok, det var der jo ikke rigtig noget at gøre ved, så jeg gik hen til Hertz for at få 'min' bil. Yes Sir, can I see your credit-card please?? blev jeg spurgt. På det tidspunkt havde jeg ikke noget kreditkort, så det kunne jeg ikke rigtig klare. Damen så virkelig ud til at være ked af det, men så klarede hun op. Er du fra IBM??? Joh, det var jeg da. Om jeg kunne bevise det?? Heldigvis havde jeg da min IBM batch, så det var ikke noget problem. Så fik jeg bilen mod at erlægge et depositum på 500\$. Med andre ord, en IBM batch er næsten lige så god som et kreditkort.

Bilen jeg fik udleveret havde automatisk gear - naturligvis. Det havde jeg ikke prøvet før, så jeg manglede naturligvis en pedal der nede i bunden. Det resulterede i at jeg trådte bremsen helt ned, da jeg ville prøve at skifte gear, jeg plejer jo at skulle koble ud. Jeg skal lige love for den kunne bremse. Det var tæt ved at den forreste kofanger bankede ned i asfalten. Der var et par fornøjede 'tilskuere', og de så ud som om det ikke var første gang de så sådan en opbremsning.

Jeg satte så kurs mod hotellet som jeg fandt uden alt for meget besvær. Det første damen i receptionen bad om var mit kreditkort! Også hende måtte jeg naturligvis skuffe. Jamen, sagde damen, så måtte jeg da betale forud. Ah, er du fra IBM?? Joh, efter hun havde set min batch kunne jeg flytte ind uden yderligere komplikationer. Jeg begyndte faktisk at blive rigtig glad for det lille skilt.

På hotelværelset så jeg, som det første, at der i et lille bæger var en engangstandbørste og en engangs barberskraber. Den eneste gang, jeg har set sådan noget på et hotelværelse, faldt sammen med den eneste gang jeg havde brug for det. Det er altså ikke altid statiske usandsynligheder dækker over noget ubehageligt. Jeg har senere boet på samme hotel, og da var der hverken tandbørste eller barberskraber, men til gengæld havde jeg min bagage.

Det var en stor oplevelse at prøve at arbejde i Pookie. Pookie var jo 'stedet', når det drejede sig om VM. Det var byen alle talte om, så den skulle man naturligvis se

for at være en *rigtig* VM'er. Nu er en IBM lokation én ting, en by noget ganske andet, og som by betragtet er Poughkeepsie bestemt ikke noget at skrive hjem om, der var nogle veje, et par indkøbscentre, mange restauranter og spredt bebyggelse . Vi var seks mand, der fik til opgave at skrive en bog. To kom fra Australien, én fra Filippinerne, én fra Sydafrika, og én fra Skotland og så var der lille mig fra Danmark. Projekt lederen kom i øvrigt fra Chile. Det var altså en ret broget skare, så det var en glædelig overraskelse at vi rent faktisk kunne arbejde sammen. Vi havde naturligvis diskussioner, men de alvorligste havde vist nok deres udspring i sproglige problemer. Når to personer fra samme land og med samme modersmål, kan have problemer med at kommunikere, hvordan er det så ikke for to personer, der taler sammen på et engelsk, som ingen af dem er for gode til???

Ian fra Australien havde næsten samme baggrund som jeg (han var bare meget yngre og flottere), altså som systemprogrammør. Vi blev hurtigt enige om at vi i fællesskab skulle skrive de mere tekniske dele af bogen, og da han var bedst til engelsk (de mener faktisk de taler engelsk i Australien) og hurtigt på en skrivemaskine (det påstod han), blev det altså ham, der skrev således at jeg hovedsageligt dikterede. Dikterede er naturligvis ikke det rigtige udtryk, det var jo ikke mig, der skulle forfatte bogen. Vi fik en masse frugtbare diskussioner undervejs - heldigvis blev det kun til diskussioner, han var *meget* større end jeg. Det *var* naturligvis ikke en skrivemaskine, men en EDB terminal, vi skrev på.

Når man skriver en bog i IBM, en bog som også skal læses af kunder, er det vigtigt at sproget er *helt* korrekt, det gælder stavning ordvalg, grammatik og sammensætning - og nåh ja, indholdet er måske heller ikke helt uvæsentligt, men det kommer måske i anden række? Derfor skulle *alt* gennemgås af en 'editor'. Ian havde jo engelsk som sit modersmål, nogen vil nok mene at det er en mærkelig afart af engelsk, men dog engelsk, så han havde ikke noget problem med at sende vore udgydelser til revision. Han blev *meget* klogere! For det første er amerikansk ikke det samme som engelsk - slet ikke, men heller ikke hans sprogbrug kunne altid accepteres, så når vort dokument kom retur, var det næsten altid overmalet med røde kommentarer. Ian følte sig ydmyget, og han var ind imellem rasende, egentlig lidt mærkeligt, for jeg rettede flere gange på hans måde at formulere sig uden han brokkede sig

Jeg havde en naiv opfattelse af, at når jeg nu var kommet til VM's tekniske højborg, var der nok en eller anden skrap fyr, der skulle revidere/kontrollere vort arbejde, specielt med henblik på at gøre det tekniske korrekt. Det gik dog snart op for mig, at specialisten, der skulle checke at vores arbejde var i orden - teknisk set, nok nærmest var mig selv - fordi jeg var den mest erfarne. Der var i hvertfald ingen andre, der gjorde noget i den retning, hverken fra vort hold eller fra centeret. Det var en pudsig erkendelse, men egentlig ikke så mærkelig. Staben i sådan et teknisk

center, var naturligvis IBM'ere ganske som jeg selv, så hvorfor skulle de være bedre/dygtigere end mig???

Vor opgave var at skrive en bog, der skulle hjælpe små kunder med et VSE system, med at løse de nye opgaver, der kom, når de udover VSE også fik VM. Det var jo netop den slags kunder jeg havde hjemme og som jeg gav assistance, så det var lige mig, men selvfølgelig var projektet jo netop også lavet for at skaffe mig en tur til Pookie. Vi fik skrevet bogen, 'Administrators Guide to VM/SP', og den blev vistnok en succes. Jeg ved at IBM-skolen i Danmark udleverede den til eleverne på alle de kurser, hvor den var relevant. Desværre er der ikke noget honorar til forfatterne - altså udover den sædvanlige løn og selve turen, men det er vel heller ikke helt dårligt.

VM laboratoriet i Endicott var i gang med at skrive en tilsvarende bog. Den skulle være en del af VM's standard bibliotek. Endicott var interesseret i at sammenligne vore idéer og tanker med deres, så vi blev inviteret en tur til Endicott. Endicott må være en af de mest udprægede IBM-byer, der findes. Næsten alle husene bar skilte med IBMs logo, parkeringspladser var enten for IBMere, IBM-kunder eller for IBMere og IBM-kunder. Det var ikke helt let at finde frem til dem, vi skulle tale med. Men vi fandt endelig en hovedreception, hvorfra man ringede til rette vedkommende, der boede i et kontor, der lå et par blokke væk.

Nu er jeg nok ret naiv af naturen, så jeg blev ret skuffet. Jeg havde en fornemmelse af at når vi fra felten fik lejlighed til at mødes med udviklere fra elfenbenstårnet, skulle vi udveksle meninger. Hvad jeg mener er at det burde være rart for disse folk, der sidder så langt væk fra den virkelige verden, at høre lidt om de problemer, vi slås med i felten, når vi omgås med de kunder, firmaet lever af. Men nej, mødet gik med at Endicott-folkene fortalte os, hvor utroligt dygtige de var, og det havde vi nok vor egen mening om i forvejen, rigtig dygtige folk har altid tid til at lytte, det er bl.a. nok det, der har gjort dem dygtige!!!! Jeg følte at Endicott-folkene måske prøvede at påvirke os til at gøre som *de* syntes, hvor *jeg* mener at de skulle lytte til den erfaring de nu havde mulighed for at lære af.

På trods af et par små skuffelser, må jeg dog indrømme at det var en meget stor oplevelse at se 'Guds eget land' - naturligvis. Jeg var i New York et par gange. Jeg ved godt jeg ikke fik set ret meget, det gjorde ikke så meget, for jeg tror ikke jeg vil kunne vænne mig til at bo der, men selvfølgelig er der meget jeg gerne ville have set, hvis der havde været mere tid. Vi 'besteg' et par af de kendteste syskrabere (Empire State og World Trade Center), men heller ikke fra oven synes jeg New York er speciel smuk, men dog meget imponerende. En af fyrene fra Australien udtrykte det meget præcist med ordet *UNREAL* altså uvirkeligt. Det var lidt af en skuffelse at frihedsgudinden var under reparation, så hun var helt gemt under stilladser og slet ikke til at se. Derimod var området lige nord for New York til min store forbavselse utrolig smukt. Hudson floden minder en del om Rhinen, men den

er naturligvis bare *meget* større - som alt andet i 'Staterne', og så er der naturligvis ingen gamle middelalderborge. Floden er omkranset af lave bjerge, og med de smukkeste efterårsfarver, er det lige ved at være *for* meget. Mådehold *er* ikke amerikanernes stærke side. Flyveturen hjem fra Pookie til New York var i øvrigt også lidt af en oplevelse. Vi fløj langs med floden i et utroligt flot vejr, og da vi nærmede os den store by, havde vi en meget flot udsigt til Manhattan skyline. Det var virkelig imponerende at se alle disse skyskrabere fra denne vinkel.

På en weekendtur besøgte vi militærakademiet Westpoint, nok USA's mest berømte officersskole. Der var mange imponerende ting at se, og højt oppe på en bjergtop fandt vi et lille museum, der var et fort fra den tid hvor Washington sloges med englænderne - altså uafhængighedskrigen. Vi så blandt andet et par led af den kæde man i sin tid brugte til at spærre floden for den engelske flåde, og de var til orientering temmelig store.

Nu kan man jo ikke forvente at få sin løn for at more sig 'Over there', så efter de planlagte seks uger vendte jeg hjem til hverdagen igen. Der var masser af arbejde. Og det var næsten alt sammen positive oplevelser. Der var en kunde, der havde købt en ny stor maskine, der skulle køre VM og VSE - altså 'ganske som vi plejer', men denne kunde ville benytte CMS til programudvikling, noget jeg syntes var en glimrende idé. CMS er hurtigere og mindre ressource krævende end de tilsvarende VSE værktøjer. Der var bare den hage ved CMS, at man ikke kan dele program biblioteker, sådan at flere kan læse fra og skrive til de samme biblioteker samtidig uden risiko for at ødelægge bibliotekernes data, og netop *det*, insisterede kunden på at kunne, for det kunne han under VSE. Det blev lidt af en udfordring, men jeg gik i gang. Jeg har altid haft det fint med min familie, så de ville ikke blive begejstret for alt for meget overarbejde, og da det var længe før man fik PCere eller skærm-terminaler stående hjemme, blev programmerne skrevet på en skriveblok - meget af det medens jeg sad hjemme på altanen og i øvrigt nød skønne sommeraftener. Det må have været en dejlig sommer, for jeg husker at jeg tilbragte mange dejlige aftener på altanen med en skriveblok i skødet. Dagen efter, kunne jeg så tilbringe et par timer med at knække mine udgydelser ind i VM - nok ikke den mest effektive måde at arbejde på, men jeg nåede det ønskede resultat. Jeg 'opfandt' en service maskine - alting i VM sker jo i virtuelle maskiner - som brugerne kunne kommunikere med. Denne service maskine 'ejede' alle bibliotekerne. Der var sikkerhed for at kun de rette personer fik lov til at benytte disse biblioteker og at andre ikke kunne - sikkerhed har altid været vigtig for mig, meget mere end det var for mine 'kunder'. Systemet havde kommandoer, til at sende 'meddelelser' mellem brugeren og service-maskinen, og sådan en kommunikation er ikke hel triviell - specielt ikke, da alting blev skrevet i det der primitive gammeldags exec-sprog (exec2). Jeg fik det til at virke, det blev testet meget grundigt, og jeg fik endog skrevet en grundig beskrivelse - dokumentation er et arbejde computer-folk som regel ikke er særlig

begeistret for, og jeg er nok ingen undtagelse i den henseende, men det blev altså gjort.

Det meste af dette system blev lavet i min fritid, medens min chef var på ferie. Da han kom hjem lagde jeg beskrivelsen af systemet på hans skrivebord og bad ham gennemlæse den. Jeg skal lige love for han gjorde store øjne, jeg tror faktisk han var ret imponeret. Men målet blev nået, kunden fik sin maskine, VM og altså mit bibliotekssystem. Også dette blev en hel pæn succes. Den pågældende kunde havde ikke betalt os for at skrive dette system, men for at få adgang til dets faciliteter, så systemet tilhørte IBM. Derfor begyndte vi at fortælle vore andre kunder om det, og jeg tror der var mere end 10, der var så interesserede, at de betalte for at få det. En morsom ting var at en pige, der skiftede job fra en kunde, der havde 'mit' bibliotekssystem, til en kunde, der *ikke* havde det, insisterede på at få det. Hun syntes ikke hun kunne leve uden - jeg var vistnok (helt bestemt) *lidt* stolt, specielt da det var flere år efter systemet var udviklet, og der i mellemtiden var kommet langt bedre værktøjer til at skrive sådan et system, så det på det tidspunkt kunne laves langt mere elegant og vel også enklere, men penge i IBMs kasse er altid en god ting - specielt, når man arbejder i en salgsafdeling.

Jeg ved ikke om det startede med denne uhensigtsmæssige måde at arbejde på, men jeg fik hurtigt et brændende ønske om at kunne arbejde hjemme ved en terminal. Hvorfor? Joh hjemmearbejde betyder dels at man kan være mere sammen med familien, men det betyder også at man kan arbejde, når man er i humør til det eller lige når man får den gode idé. Dengang var det bare ikke praktisk muligt. Terminalerne var dumme (PC'erne var endnu ikke almindelige, de var meget dyre - måske var det også før de kom på markedet), så der krævedes en kontrolenhed, der var *alt* for stor til vores beskedne lejlighed - men så kunne den selvfølgelig også klare 7 terminaler, hvad jeg naturligvis ikke ville have nogen glæde af!!! Det blev altså ved drømmen - indtil videre.

Der var én af mine VM installationer, der var lidt ud over det sædvanlige. Der var solgt en lille maskine til Dansk Hydraulisk Institut, DHI. Nu sker det ofte at kunden helst vil have maskinen leveret lige før kontrakten er underskrevet. *Når man endelig har besluttet sig, så skal det også være*, så det blev til en hastesag. Det lykkedes at skaffe alle komponenterne - bortset fra tapestationer. Det lyder måske ikke så alvorligt, men da alle system programmerne blev leveret på tapes, var det faktisk en alvorlig forhindring, for ikke at sige at det gjorde installationen helt umulig - tilsyneladende. Man fandt dog ud af (nej, det var desværre ikke min idé) at det måtte være muligt at køre diskene ind til mig og koble dem på min lille 'private' VM maskine (det *var* naturligvis afdelingens, men da jeg var VM manden, var det i praksis min maskine). Jeg kunne så installere VM på disse diske, der derefter kunne køres til Hørsholm og kobles på DHI's maskine. I VM kan man definere en virtuel maskine, der er magen til *den* kunden skal have. I denne maskine laver man så sit system, og hvis det kan køre i den virtuelle maskine, kan det som regel også

køre på den rigtige (det *er* dog sket at en virtuel maskine på visse områder har været bedre end en virkelig). Teknikeren fik at vide at hvis han slettede diskenes indhold under maskininstallationen, ville han få lov til at bære dem på ryggen - fra Hørsholm ind til Vester Søgade, så vi kunne genskabe systemet. Det lykkedes!!! Kunden fik sit system, og tapestationerne ankom, før der blev det store behov for dem. Jeg tror nu nok vi var lidt mere omhyggelig med planlægningen, end normalt, det er jo ikke så morsomt at flytte diske frem og tilbage mellem Hørsholm og København, fordi man ikke har tænkt sig om godt nok. I øvrigt var det en kunde, som jeg aldrig senere fik besøgt, så måske er de stadig ikke klar over at jeg var medvirkende årsag til at de kunne få deres maskine så hurtigt. DHI var dog ikke hel ukendte for mig, jeg havde haft kontakt med dem et par gange medens jeg hjalp DTU med deres installation.

I disse hektiske dage tror jeg vi fik sat en rekord, der hører hjemme i den store rekordbog. Et rejsebureau havde gevaldige kapacitetsproblemer, deres maskine var simpelthen *alt* for lille. Deres flinke sælger gik så i gang med at hjælpe dem (det er jo *det* sælgerne tjener deres penge ved). Han fandt en passende maskine, som blev installeret samme uge, som den blev solgt!!!! Denne maskine indeholdt i øvrigt noget helt nyt for mig. Den havde to processorer, noget som VM ikke kunne håndtere særlig godt uden man gjorde en indsats for det. Det gav mig imidlertid lejlighed til at lære lidt om multiprocessorer, en viden jeg og mine kunder bestemt har haft rigtig megen glæde mange gange af senere.

En kunde fik en fredag nat besøg af meget uvenlige personer. De kunne let komme ind i maskinstuen fra bygningens parkeringsdæk. De medbragte en antal lægte-hammere (det er dem tømmerne bruger, dem der har en stor grim spids i den ene ende, så man kan hugge dem ind i en bjælke), og nogle glas med kongevand. Kongevand er en blanding af saltsyre og salpetersyre. Det er en syre så stærk at den kan opløse guld. Med hamrene lavede man nu huller i maskinerne og hældte derefter kongevand ned i hullerne. *Det* er der ikke mange maskiner, der kan holde til, og heller ikke disse. Uheldigvis for kunden, var der ikke særligt gode backup-tapes, så der var rent ud sagt et grimt problem. Manglende back-ups var vist desværre ikke et specielt problem for denne kunde, mange af vore kunder havde vist al for megen tiltro til maskinernes stabilitet, så megen at de ikke gjorde ret meget ud af sikkerhedsprocedurerne, det gælder både for systemet og adgangen til maskinerne. Det er egentlig mærkeligt for kunderne var naturligvis meget afhængige af deres data og vi havde omhyggeligt indprentet dem vigtigheden af sikkerhed, men altså ikke nok!

Et større rejsebureau havde en mindre tilsvarende maskine til overs - den var vist netop blevet erstattet af en større og kraftigere. *Den* 'lånede' IBM. Teknikerne fjernede så pladehukommelserne fra de beskadigede diske, og satte dem ind i en frisk boks og kopierede indholdet til en helt ny pakke. Det lykkedes. Man kunne

læse pladerne én gang (ingen prøvede om de kunne læses flere gange), så kunden fik reddet sine data. Den følgende onsdag kunne man starte produktionskørsler på den 'nye' maskine, noget jeg tror må være tæt ved en verdensrekord. Selv havde jeg ikke noget med denne sag at gøre, men kunden var en VM-kunde, som jeg ofte havde besøgt, og som jeg har haft kontakt med mange gange siden, og jeg har set, hvordan tapeskabene var ræverøde af rust, på grund af ætsningen fra syredampene. Disse skabe stod ikke i maskinstuen, men i et kontor på den anden side af gangen, det var altså virkelig stærke sager der var anvendt.

Hvorfor nu det?? Joh, det er jo svært at vide når man ikke har fundet de skyldige, men den almindelige opfattelse var, at det nok skyldtes at den pågældende multi-nationale kunde havde kontakt og samhandel med Sydafrika, og på *den* tid var Sydafrika meget upopulær på grund af apartheid.

I øvrigt blev maskinerne derefter placeret i et rum uden vinduer, et rum, der er låst med en kodelås, og sådan bør det naturligvis altid være med maskiner, der indeholder vigtige data, og for de allerfleste firmaer er alle data meget vigtige.

Det var ikke alle IBMs salgsafdelinger, der var så godt forsynet med VM ekspertise som min. Da der blev solgt en VM-maskine til Universitetets geografiske institut, blev jeg derfor udlånt for at assistere med installationen. Den IBM'er, der stod for projektet var specialist i MVS og APL, men kendte ikke noget eller kun meget lidt til VM, så han havde behov for hjælp til installationen og uddannelse af universitetets folk. Det blev en af mine mest vellykkede installationer. Jeg kom en morgen (altså ved nitiden) og begyndte installationen. Allerede ved frokosttid havde vi et kørende system. *Det* var noget, der imponerede folkene fra universitetet, de havde haft konsulenter fra andre firmaer, der havde brugt ganske anderledes lang tid (vistnok uger) på at få *deres* systemer i luften. De følgende dage gik med at uddanne folkene fra universitetet og IBM konsulenten. Det var ret morsomt, for IBM'eren var den tidligere leder af matematisk afdeling, altså ham, der i sin tid havde *de* 1620 kunder, jeg passede nogle (mange) år før, så vi kendte altså hinanden vældig godt.

En dag noget senere ringede han og spurgte om jeg havde mulighed for at give noget REXX uddannelse på instituttet. REXX uddannelse havde jeg givet mange gange, så jeg havde naturligvis en stak foils, der var parate til at blive vist frem for hvem som helst, der kunne tænkes at være interesseret. Jeg drog ind på universitetet, hvor der i et rum sad en hel del venlige (og forventningsfulde?) personer. Jeg vendte mine foils, og snakkede til dem, som det sig hør og bør. Da jeg havde holdt mange REXX kurser, var det slet ikke noget problem for mig, som var og er vældig glad for REXX. Bagefter fik jeg så at vide at de folk, jeg havde underholdt, var lærere, der selv skulle holde REXX kurser. Se *det* ville jeg egentlig godt have vidst i forvejen. Men ingen beklagede sig. De var tvært imod så venlige at fortælle at min seance havde struktureret REXX for dem på en ny måde, som de nok mente de kunne få glæde af. Jeg tror nu nok jeg havde talt en hel del

anderledes, hvis jeg havde vidst at det var et kursus for lærere, men måske var det en fordel for mig at jeg ikke vidste noget, det er da muligt jeg ville have følt mig lettere presset af situationen, der jo var anderledes end det jeg var vant til.

En af IBM's re-organiseringer indebar at 'min' afdeling 'Systems Marketing' blev slået sammen med en langt større afdeling under navnet 'Industri Sektoren'. Afdelingerne havde før heddet 'Branch Offices' men nu blev de altså større og blev derfor kaldt sektorer. Det var slet ikke dårligt, set fra mit synspunkt. Det betød nemlig at jeg nu fik mange flere kunder, og nu var det ikke mere nødvendigvis små kunder. Store kunder, der selv havde systemprogrammører, gav ganske anderledes udfordringer i form af mere kvalificerede problemer, og måske især mere VM relaterede problemer. Ydermere blev mit kundeområde udvidet, så jeg fik en hel del kunder i Jylland. Det betød naturligvis mere arbejde og flere rejser, men det gjorde slet ikke noget, fordi det var et arbejde som jeg holdt umådelig meget af. Jeg havde ingen faste kunder, min opgave var at sørge for at sektorens mange VM-kunder fik alle deres VM problemer løst på bedste måde, og naturligvis at forsøge at få dem til at købe flere og større maskiner, men vigtigst af alt, jeg skulle sørge for at kunderne hele tiden var tilfredse med deres VM system. Da jeg altid har været på fast løn (eller hvad man nu vil kalde det), var salget ikke direkte vigtigt for mig, jeg syntes der var vigtigere - og sjovere - at sørge for at de installerede maskiner blev udnyttet så effektivt som muligt, altså at hjælpe kunderne med at undgå at købe nye maskiner før det var absolut nødvendigt. Jeg tror dog at denne attitude var en fordel for alle parter. Kunderne kunne jo stole på at jeg hjalp dem, så godt, som jeg kunne, jeg ville jo overbevise dem om at VM var (er) et pragtfuldt og meget effektivt system. Sælgerne var måske mere interesserede i bare at sælge (der var måske en ny bil eller et hus, der skulle betales??). Men hvis kunderne blev ved med at være tilfredse, var det nu nok også mere sandsynlig at, de blev ved med at være kunder.

Naturligvis brugte jeg en hel del af min tid med at fortælle kunderne om VM's fortræffeligheder, og dertil skulle jeg naturligvis bruge foils. Jeg brugte foils som disposition for mine indlæg mere end for at kunderne skulle have noget at kigge på, men jeg ville da gerne have mine foils til at se pæne ud.

Derfor gik jeg i gang med at lave nogle SCRIPT skeletter, så jeg letterne og mere ensartet kunne producere disse små plastikstykker. Da jeg gerne ville have teksten til at stå på midten af foil'en, fandt jeg en SCRIPT kommando, der påstod den kunne justere vertikalt - altså lige det jeg havde brug for. Jeg kunne bare ikke få den til at virke. Det skulle dog ikke være noget stort problem, for på den tid delte jeg kontor med landets SCRIPT ekspert, altså ham der var for Script hvad jeg var for VM. Selvfølgelig konsulterede jeg ham, men han kunne egentlig ikke rigtig hjælpe, han kendte selvfølgelig kommandoen, men han mente at have hørt at den ikke rigtig virkede.

En dag beklagede jeg så min nød til Jon. Endnu en gang gik det op for mig at han tilsyneladende må være alvidende. Uden at blinke med øjnene, sagde han: "Åh ja, det problem kender jeg, du skal bare".

Naturligvis havde han ret, hvordan han kunne slå SCRIPT specialisten på 'hjemmebane', forstår jeg stadig ikke, men jeg var/er *meget* imponeret.

Kort tid efter jeg kom til Industrisektoren, havde jeg været i IBM i 25 år. Sådan noget bliver naturligvis fejret. Min afdeling havde fået en ny sekretær, der med stor iver kastede sig over den meget vigtige opgave, at arrangere mit jubilæum - det var noget hun ikke havde prøvet før, men hun klarede opgaven med bravour.

Sådan en dag starter normalt med at man bliver afhentet i sit hjem af en fornem limousine - naturligvis skal fru'en med. Man ankommer så til afdelingen, hvor venlige (den dag er de alle utroligt venlige) og glade kolleger modtager én med viftende flag og hurraråb. Derpå er der morgen kaffe i afdelingen - måske vanker der et par gaver. Derefter er *der så* reception, hvor kolleger og evt. kunder (der kom nu ingen kunder til mit jubilæum - men dog et par hilsener) kan hylde jubilaren - og måske overrække et par gaver. Endelig er der frokost for en halv snes udvalgte kolleger, den nærmeste chef og naturligvis fru'en. Ved denne frokost blev der endnu en gang overrakt gaver, dels en fra frokostgæsterne, men også fra IBM, der på de tider insisterede på at give et ur, en check på 10.000 Kr. og til knaphullet et lille guldemblem, der viser at man er medlem af den yderst eksklusive 'Quarter Century Club' (QCC). Det blev en dejlig dag. Jeg har været med til adskillige jubilæer, men mit eget var naturligvis det bedste. Vi fik en vældig god frokost på Dronning Louises kro i Nivå - jeg synes man så vidt muligt skal støtte de lokale. Det er særdeles hyggelig at sludre om gode gamle dage med nogle af de mennesker, der var vigtige for én, i starten af éns karriere, men det bedste var nu nok alle de positive tilkendegivelser fra mange kolleger. Jeg var nok heldig med at være VM specialist, det gav mig jo mulighed for at hjælpe en mængde af mine kolleger med råd og dåd, og jeg husker da tydeligt at Kirsten var ret imponeret, da vi kom hjem. Hun troede rent faktisk at alle *mente* alle de pæne ord, hvad de måske/forhåbentlig gjorde. Det værste ved sådan et jubilæum er nu, at man dagen efter skal prøve at vænne sig til at være 'et almindeligt' menneske igen og det kan da godt være lidt svært.

Gennem hele dagen blev vi ledsaget af en fotograf, så et par dage efter jubilæet fik jeg et lille album som et varigt minde om dagen. Min sekretær var meget skuffet, det var 'lykkedes' hende at undgå fotografen, så hun ikke var med på et eneste af billederne. Jeg havde nu også gerne set at hun var med på et eller flere af billederne, hun havde jo stået for hele arrangementet, så hun var da en vigtig del af dagen og derfor burde hun også være i mindebogen.

En af fordelene ved at invitere kolleger til sit 25 års jubilæum er, at de så måske husker at invitere én til *deres* jubilæum. Det lykkes for mig, jeg har faktisk været med til en hel del af disse jubilæer, og de har alle været *meget* hyggelige. Den

første jubilæumsinvitation jeg fik blev jeg meget glad for, det var en kollega helt tilbage fra 1620 tiden. Naturligvis skyndte jeg mig at takke ja, og så gik der kun et par dage før jeg blev inviteret til endnu et jubilæum - på samme dag. Jeg var glad for at jeg ikke fik disse to invitationer samtidig, i så fald ville jeg jo skulle beslutte hvilket jeg skulle gå til, nu havde jeg sagt ja tak til det første, og kunne derfor beklage til den anden at jeg desværre var optaget.

Ét af disse jubilæer var lidt specielt. Vi - frokostgæsterne - fik besked på hvor og hvornår vi skulle mødes, men vi kunne ikke få at vide, hvor vi skulle hen. Vi kørte ind mod byen og begyndte at gætte. Efter retningen troede vi, vi skulle til Lumskebugten, men nej vi fortsatte til Nyhavn. Der er jo også steder, man kan spise og spise godt, men heller ikke det. Vi blev sendt om bord på en lille motorbåd, der sejlede os rundt i havnen på en rigtig hyggelig havnerundfart. Der var laksemadder, øl og hvidvin, så det var rigtig rart. Efter denne tur blev vi så endeligt sendt i land - netop ved Lumskebugten, hvor vi fik den sædvanlige gode frokost.

I de dage var der et par ekstra fordele ved at have været mere end 25 år i IBM. Dels fik man en ekstra uges ferie om året - fremover. Dette gjaldt dog kun for IBM Danmark, og skyldtes vistnok en oversættelsesfejl. Men det gjorde ikke noget, en ekstra uges ferie er og bliver en ekstra uges ferie. Desuden holdt man én gang om året en middag for QCC medlemmerne, og disse middage var bestemt en oplevelse.

Den bedste af disse QCC middage foregik i Tivoli. Vi fik på forhånd tilsendt billetter til Tivoli og billetter til Glassalen, hvor man *det* år opførte Linie 3's 10 års jubilæums show. Vi kunne altså gå i Tivoli, når vi havde lyst, blot skulle vi være på Nimb kl. 18. På Nimb fik vi serveret pindemadder og et par glas hvidvin. En time senere gik vi til glassalen, hvor vi morede os over Linie3's flotte show. Endelig var der buffet på Nimb, hvorefter vi for egen regning kunne gå i Tivoli. Jeg tror nu ikke der var meget andet end fyrværkeriet tilbage Tivoli lukker jo kl. 12, men alt i alt en pragtfuld aften også fordi det var en af de rigtig dejlige sommeraftener.

Desværre blev IBM senere så fattig, at disse fordele blev sparet væk. Jeg har aldrig rigtigt lært at forstå, hvorfor man ikke kunne give QCC'erne en ekstra uges ferie. Mange af os havde arbejde af en karakter, der gjorde at man efter ferien måtte indhente, hvad man havde forsømt, medens man var væk, det galt i hvertfald mig, jeg havde bestemt ingen afløser, så det kunne vel næppe koste IBM særlig mange penge med lidt ekstra ferie.

Mange af kundebesøgene i Jylland var hyggelige. Jeg fik en fornemmelse af at man var glad for at jeg 'gad' drage helt fra København til Jylland for at hjælpe. De lokale (Københavnske) kunder var jo vant til at se mig meget hyppigere. Når *de* havde problemer, kunne jeg jo bare springe ned i bilen og smutte ud til dem. Jyllandsturene indebar derimod det problem at man som regel skulle gevaldigt tidligt op for at nå en flyver i Kastrup - der er som bekendt ret langt fra Kokkedal til Kastrup. To gange på én uge blev jeg sendt med 'pyjamas flyveren', derefter svor

jeg en dyr og *næsten* hellig ed på at det ville jeg ikke mere. Jeg tror nu nok jeg har været med en 'pyjamas flyver' et par gange siden, men principper har man vel for at kunne bryde dem.

Jeg husker en gang en sekretær skulle bestille en flyvebillet til mig (formodentlig til Tirstrup). Uden at tænke særlig over det, stillede hun mig det sædvanlige spørgsmål: "Ska' du ha' valuta?"

En gang tog jeg til Randers for at fortælle en kunde om hvad han kunne opnå, ved at køre VM/HPO i stedet for blot VM. HPO (**H**igh **P**erformance **O**ption) var et system man kunne installere oven på VM, med det resultat at den kunne udnytte større maskiner og især maskiner med mere end 16 MB hukommelse og 2 processorer mere effektivt. Jeg brugte en hel dag. Jeg fortalte og fortalte. VM/HPO var et af mine yndlingsemner, og jeg vidste hvad jeg talte om. Vi fik en god frokost midtvejs, og jeg tog hjem, med en fornemmelse af at have gjort et virkelig godt stykke arbejde. Et par dage efter modtog jeg en note fra den jyske sælger, en note der var så fuld af ros, at jeg næsten blev helt flov - og kunden havde været særdeles begejstret så alt syntes at være den rene idyl. Der gik så ca. et halvt år, så blev jeg kaldt til Århus på 'brandudrykning'. Jeg tog af sted sidst på eftermiddagen. Problemet var at den selv samme kunde havde besluttet sig til at købe en ny maskine hos en konkurrent - bestemt ikke noget man er glad for i IBM. Da sandheden kom frem, viste det sig at Århus afdelingen helt havde forsømt den pågældende kunde siden jeg havde været der, og det havde konkurrenten åbenbart ikke. Vi var to mand fra København, der skulle forsøge at redde stumperne, desværre var det *alt* for sent, toget var kørt, kunden havde truffet sin beslutning så den sag blev altså tabt på gulvet. Joh, det er sandelig rigtigt, man kan ikke vinde hver gang, men vi var da et par 'københavnere' der var godt sure.

Det var naturligvis ikke kun kunder, der kunne have behov for hjælp. Mine gode kolleger havde da også brug for lidt VM-ekspertise ind imellem. Jeg husker én, der havde en fil, der blandt andet indeholdt et felt med navn - fornavn og efternavn. Jeg husker ikke hvilken rækkefølge, men han skulle bytte om på fornavn(ene) og efternavn. Nu er der jo det at nogle mennesker har flere fornavne, så han havde lavet noget kodning, hvor han havde flere skærbilleder med tests. Hvis . . . og hvis . . . Sådan noget kan godt være svært at overskue, så da han ikke kunne få det til at virke, brugte han den gode gamle metode: Man finder en eller anden tilfældig stakkel, så prøver man at forklare *ham* hvad man havde tænkt sig. Det medfører mange gange at man selv kan se, hvad man har gjort forkert.

Det er i øvrigt ikke nogen helt ringe metode, og jeg har da ofte selv benyttet den. En eller anden blev altså mig. Da jeg havde hørt lidt på ham, sagde jeg at nu gad jeg høre ikke mere af det sludder. Så skrev jeg én linie kodning, der gjorde alt hvad han ville - og på en måde så alle kunne se og forstå. Han var imponeret og jeg lidt stolt.

Gode programmer skal ikke bare virke og gerne hurtigt, de skal også helst skrives på en måde så andre kan finde ud af, hvad der foregår, hvis der senere skulle komme behov for rettelser eller modifikationer. Det er jo ikke altid den oprindelige programmør er til rådighed, når programmet måske meget senere skal ændres.

Samme kollega havde et andet problem, der gjorde at vi måtte dybt ned i bøgerne, der beskriver maskinens virkemåde i detaljer. Det var et database program, der tidsstemplede rekords - simpelthen ved at kopiere maskinens binære 'ur'. Hans problem var så at få dette binære tal oversat til dato og klokkeslæt. Det var ikke simpelt. 'Uret' talte med en eller anden frekvens, startende 1/1 1900 og omkring 1972 løb tælleren fuld og startede forfra. Det lykkedes - men så sandelig ikke uden besvær. Yderligere var der noget med nogle tal i 32-tal systemet, han skulle have oversat til 10-tal systemet, men også det blev klaret med et lille elegant REXX program.

Nu var Jylland ikke det eneste sted, der blev rejst til. Der var til stadighed kurser og seminarer i det store udland, og da jeg var den lokale specialist, skulle jeg naturligvis deltage. Det gav anledning til mange gode oplevelser, og det var ikke kun EDB, der blev talt om. Jeg husker at Jon og jeg havde en gevaldig diskussion efter en tur til Rom. Det var "Piazza Del Popolo", det drejede sig om. Vi var begyndt at diskutere hvorfra navnet kom. Jeg mente at have set at det havde noget at gøre med nogle popler i en fjern fortid. Disse popler skulle have stået på pladsen for mange år siden. Jon mente at det var 'folkets' plads. Da vi var/er grundige, begyndte vi naturligvis at konsultere venner og bekendte, der måtte formodes at vide noget om det. VNET var selvfølgelig vor måde at kommunikere på, og der kom mange gode og fornøjelige forslag. Det var (naturligvis) en italiener, der kom med den mest sandsynlige forklaring. Han fortalte at tvillingekirkerne, der ligger på pladsen var bygget for midler, der stammede fra en folkeindsamling. Derfor kom sognet til at hedde folkets sogn, og pladsen blev så opkaldt efter sognet. Derfor!!!!!!!!!!!!

En morgen skulle jeg til Paris til et VM møde. Jeg havde omhyggeligt beregnet, hvornår jeg skulle køre fra Kokkedal for at have god tid - jeg hader at være sent på den. I nattens løb sendte vejrguderne en smukt lille lag sne ned over Sjælland. Det betød at trafikken på Helsingør motorvejen brød totalt sammen, så jeg kom til Kastrup *meget* sent. Jeg har aldrig set så mange mennesker i afgangshallen i Kastrup. Køerne syntes kilometer lange, men jeg var nødt til at smutte uden om, hvis jeg ville nå min flyver. Jeg husker tydeligt at 'damen' råbte efter mig, medens jeg løb op af trappen: *"Det er ikke sikkert at kufferten når at komme med"*. Da jeg kobberrød i hovedet af stress og anstrengelse kom ind i flyveren, så jeg at den var fuld af grinende IBM'ere, jeg forstår stadig ikke hvordan de var nået frem i god tid. Jeg husker også hvordan jeg stod i lufthavnen i Paris og ventede på min kuffert. Jeg blev mere og mere frustreret over at den ikke dukkede op, 'damen' havde jo truet med at den måske ikke nåede at komme med??? Joh, den var der godt nok. Den

kom som den aller sidste. Man skulle ellers tro at, når den kom ind i bagagerummet som den sidste, skulle den komme ud som den første. Det står der vistnok noget om i bibelen.

En behagelig afveksling i den daglige trummerum, som bestemt ikke var nogen trummerum, var IBM skolen. IBM skolen havde en række faste instruktører der passede hver deres område, men ind i mellem tilkaldte man specialister fra felten til at hjælpe med særligt krævende kurser. Det var noget jeg syntes rigtig godt om. Jeg tror nok jeg var glad for at undervise selv om jeg altid var lettere nervøs, og så var det en ny måde at mødes med kunderne og samtidig en mulighed for at se nogle af de kunder, jeg måske havde mindre kontakt med til daglig.

Naturligvis vidste Jon meget mere om VM, end jeg, men måske var det lettere for mig at tale til kunderne på et niveau, der passede til dem. Jeg husker f.eks. en systemkonsulent kom for at få hjælp med et eller andet VM problem. Det var ikke noget, jeg kunne klare, så jeg anbefalede ham at spørge Jon. Han var lige ved at græde, da han svarede: "Det *har* jeg gjort, men jeg kunne ikke forstå, hvad han sagde". Jeg måtte altså have fat i Jon for at få forklaringen af ham og så prøve at oversætte den til et mere almindeligt 'SE-dansk', hvad der naturligvis lykkedes.

To gange om året holdt skolen et morgenkaffe møde for gæsteinstruktørerne. Ved denne lejlighed fik vi foræret en 'Ølhund', et glaskrus fra Holmegårds glasværk med en indgraveret ugle. Jeg tror jeg har en syv-otte stykker. De er særdeles velegnede til en stille øl på altanen, en lun sommerdag, fordi der netop kan være en hel øl i dem.

På et kursus i Bruxelles, faldt jeg i snak med en af mine tyske bekendte fra et af de mange VM-møder, jeg har deltaget i. Vi kom til at snakke om Jon - den *store*, berømte VM'er, som *han* naturligvis også kendte. Han fortalte en morsom lille historie: Jon havde, af en eller anden grund været i Endicott, hvor de udvikler VM. I Endicott, sad der en masse fyre og skrev VM-kode. De havde alle et ansvar for et bestemt stykke kode, og de kendte en smule til den del af VM, der lå lige i nærheden af deres egen kode. Men en dag gik det op for ledelsen, at der faktisk ikke mere var nogen i Endicott, der kunne huske, hvad VM egentlig var for noget. Når man så havde besøg af Jon, kunne man jo passende lade ham lave et lynkursus om VM, for VM udviklerne!!!!

En temmelig mærkværdig historie, som jeg naturligvis måtte have bekræftet, da jeg kom hjem. Jon grinede, da jeg fortalte ham det. Jeg fik så den rigtige historie. Jon havde til eget brug udviklet en ny programmerings teknik eller måske kan man kalde det et programmeringssprog. Det var nogle assembler makros, der gjorde det muligt at skrive struktureret assembler kodning, noget der kunne øge produktiviteten hos programmører. Det hedder naturligvis PL/Jon. Jon havde fortalt om det i Endicott, og de var *så* interesserede at de inviterede ham over, for at fortælle om det. Og når han nu alligevel var i gang, kunne han da også fortælle om VM i al

almindelighed, for det *var* faktisk rigtig at man trængte til sådan en orientering - man havde glemmt, hvad der var væsentlige var ved VM - og måske mistet overblikket.

Medens jeg var i industrisektoren, kom der meddelelse om et par VM residencies i Pookie. Da jeg mente at emnerne lige var noget for mig, og da jeg havde lyst til at se 'det store land' endnu en gang, fik jeg min chef til at tilmelde mig. Jeg blev afvist begge gange. Det fik min chef til at spørge Pookie om det var mig eller Danmark man ikke ville se. Svaret kom omgående. Man havde opfundet et lille ekstra job for to mand i to uger, og man ville naturligvis *meget* gerne se mig som ene af deltagerne. Pookie's problem var at det sidste residency ikke kunne lave deres arbejde færdigt fordi programopdateringerne man skulle teste ikke var kommet. Vores job var så at installere de forsinkede programmer, så man kunne finde ud af om de virkede.

Jeg kom altså til Pookeepsie endnu en gang. Denne gang skulle jeg køre i bil fra New York til Pookeepsie. Jeg må tilstå at jeg ikke var helt tryk ved at skulle køre fra Kennedy lufthavnen (der er en af verdens største) og ud af New York, der jo er en af verdens største byer med en berygtet forrygende trafik. Jeg fik 'min' bil og en hurtig forklaring. Naturligvis havde jeg studeret et kort hjemme fra, så jeg gav mig fortrøstningsfuld på vej nordpå. Det var ved White Plains, det gik galt. Jeg valgte den højre bane (jeg har altid befundet mig bedst ved vejkanterne) hvor jeg skulle have valgt den venstre. Resultatet var at jeg fik en lang tur gennem det efterårs-smukke Connecticut. Desværre gik der en hel del kilometer, før det gik op for mig at der var noget galt. Jeg kørte derfor ind på en servicestation og kontaktede en tilfældig nydelig mand - amerikanere *er* som regel meget hjælpsomme. Han forklarede (hvad jeg godt vidste) at jeg var *helt* galt på den, men jeg skulle forsætte til den tredje afkørsel, *der* ville jeg kunne vende. Så fik jeg en forklaring på, hvad jeg skulle, når jeg kom tilbage til White Plains. Det gik også galt, men nu var jeg så tæt på det kritiske punkt (og derfor meget forsigtig), så jeg fandt endelig den rette vej. Naturligvis ville jeg gerne se noget af 'guds eget land', men efter at have fløjet fra København til New York, var jeg nu ikke særlig oplagt til turistture, og jeg var da også rimelig træt, da jeg endelig ankom til Pookeepsie to - tre timer senere end jeg havde planlagt.

Jobbet i Pookeepsie var hurtig overstået. Vi skulle installere en opdatering til VM, noget vi begge havde prøvet mange gange, meeen man havde (desværre) opfundet et nyt kompliceret opdateringssytem til VM. Det var nok ikke fuldkomment, da vi skulle bruge det - det er nye programmer jo sjældent, så det gav os nogle problemer, men efter en tre fire dage var vi klar, og så var der et andet residency, der skulle prøve om det virkede - altså ikke mere vort problem.

Vi havde altså god tid!!! Det var den gang man havde genoptaget rumflyvningerne med shuttlen, efter det grimme uheld, så vi forlod IBM midt om dagen, og kørte til vort hotel for at se shuttlen komme hjem.

Min 'makker' var fra Canada, han var mere sælger end jeg (altså mere udadvendt), så han fik den strålende idé at vi skulle se IBM fabrikken. Vi kørte altså ned til fabrikken og præsenterede os som Curry fra IBM Canada og Preben fra IBM Danmark, og udbad os en guidet tour. Damen så helt fortvivlet ud, men det viste sig at hendes problem var, at det var tæt på frokost, og i USA spiser man frokost ude i byen. Hun havde altså ingen guider hjemme, men hvis vi kunne komme en time senere, ville det være *meget* bedre. Hjemme i DK har vi kun frokost en halv time, men vi spiser selvfølgelig også på IBM. Da vi vendte tilbage, blev vi henvist til en nydelig ung dame, der viste os så meget af fabrikken, som hendes badge gav hende adgang til. Det var mærkeligt at se sådan en fabrik. Jeg mener der på den tid vistnok var ca. 12000 ansatte, og vi så næsten ingen. Det meste foregik ved hjælp af robotter, der stod anbragt i store glasbure (formodentlig af hensyn til støv). Små automatiske tog kørte så rundt mellem robotterne og hentede og afleverede stumperne. Det var ret imponerende!!!

Weekend'en benyttede vi til at køre rundt i Hudson dalen. Også denne gang var det efterår med træerne i de smukkeste farver, man kan forstille sig. Curry ville gerne se IBM laboratoriet i Yorktown Heigh, så vi kørte derop. Der var et tomt portnerhus og en bom. Vi kørte op til bommen. Vi blev iagttaget af et nysgerrigt TV kamera, men ved hjælp af et samtale-anlæg, lykkedes det os få en eller anden til at åbne bommen, så vi kunne komme ind og se på huset - dog kun udefra. Det var virkelig meget flot. Der var en storslået udsigt over vidtstrakte plæner, skove (dem der med alle farverne) og Hudson floden i det fjerne.

Curry tog hjem før mig, så jeg blev alene tilbage uden alt for meget at gøre. En dag tog gutterne fra de andre residencies til frokost uden mig, ikke noget problem, jeg kan sagtens klare mig uden frokost, men en asiat fra Singapore dukkede op og spurgte om jeg ville med ham ud at spise og om jeg kunne li' kinesermad. Han kendte et sted, hvor de lavede den slags som han var vant til hjemmefra. Det var helt sjovt at se ham snakke med tjeneren på kinesisk - meget ivrigt - men resultatet blev absolut spiseværdigt, sandsynligvis det mest kinesiske jeg nogen sinde har fået og jeg *kan* altså vældig godt lide kineser mad.

Efter hånden skulle jeg så også hjem. Min flyver lettede fra New York ved nittentiden, så jeg planlagde at køre de ca. 150Km i ro og mag - og så skulle jeg i øvrigt være ude af mit værelse tidligt om formiddagen. Jeg kunne jo altid stoppe op undervejs og se mig om i den smukke Hudson-dal. Jeg er ikke altid heldig med vejret, så det styrt regnede netop den dag. Altså måtte jeg køre direkte til JFK, hvor jeg ankom først på eftermiddagen. Den dag var jeg glad for at jeg rejste på business

class, det gav mig nemlig adgang til en stille og rolig ventesal, hvor der var aviser, magasiner, kaffe og andre (bedre?) drikke varer. Den slags er virkelig en stor behagelighed, når man skal slå nogle timer ihjel alene i en dum lufthavn. Det var for øvrigt nok en af de sidste gange jeg prøvede business class, IBM var på det tidspunkt begyndt at blive meget omkostnings-bevidst.

Da jeg kom hjem, fik jeg et nydeligt brev fra Pookeepsie, hvor man takkede mig for min storslåede indsats - uden mig ville det ikke have været mulig at Joh, der er sandelig også sælgere i de tekniske afdelinger, men det kan da godt være at de mente noget af det.

Jeg blev helt gode venner med Curry, så jeg var lidt ked af at tage afsked med ham, men *han* var *helt* sikker på at vi nok skulle mødes igen. Den store filur. Jeg havde godt nok lagt mærke til at han flere gange havde været inde hos den lokale chef, men da det åbenbart ikke var for at klage over mig (vi havde det faktisk fint sammen), tog jeg mig ikke af det. Han havde været inde for at aftale et assignment der på stedet, han skulle altså arbejde fast i Pookie i et par år. Det man gør på de kanter - ud over at køre disse residencies, er at drage rundt i verden for at introducere nyhederne i VM for 'os i felten'. Jeg kom til et sådant introduktions møde i Rom, og sandelig om ikke Curry og projektlederen fra vort residency dukkede op som instruktører. Det var Curry's første besøg i Europa, og Europa er altså noget helt andet end USA og Canada, og da jeg havde været i Rom flere gange, påtog jeg mig rollen som guide - ikke noget problem, jeg holder virkelig meget af Rom. Jeg husker vi aftalte at mødes i Peterskirken - jeg var dog den eneste, der kom til den aftalte tid, de to andre havde *meget* travlt med at planlægge deres hjemrejse, de havde fundet ud af at de kunne flyve østpå, så de kom over Japan hjem. På den måde ville de jo få en meget mere spændende tur ud af det. Jeg hørte aldrig om det lykkedes for dem, men det gjorde det jo nok.

Der var en overgang, hvor sektorerne hold kick-off møder et par gange om året. Det var møder, der var beregnet til at planlægge strategier for afdelingerne, så de bedre kunne opnå deres målsætninger. Jeg var til sådan et møde i Skåne en sommer. Det var netop på Kirstens fødselsdag, så det var ikke rigtig velset at jeg tog af sted. Afdelingen havde i forsommeren haft en salgskonkurrence, hvor præmien var en tur til Rom - med ægtefælde. Heldigvis for mig, var der ikke solgt så meget, så kun få sælgere havde kvalificeret sig. Nu havde man arrangeret turen, så man valgte at sende nogle systemkonsulenter af sted i sælgernes sted. Når jeg nævner dette, er det naturligvis fordi, jeg blev en af de heldige (dygtige??) der vandt. Da man fandt ud af at det var Kirstens fødselsdag, blev jeg kommanderet til at ringe omgående for at fortælle den gode nyhed. Og joh, hun ville da gerne med!!!! Det blev en herlig tur. Kirsten havde aldrig prøvet at flyve, så hun var noget spændt. Nu er det at flyve som bekendt normalt ret udramatisk, så da Kirsten havde

prøvet det en enkelt gang, holdt hun op med at være nervøs hver gang jeg skulle ud at flyve.

Rom er altid pragtfuld. Af hensyn til skattemyndighederne, var vi IBM'ere nødt til at deltage i nogle kedelige IBM-ture - fabrikker og sådan noget. Imedens blev vore ægtefæller kørt rundt i Rom på 'rigtige' turist ture. Kirsten fortalte med begejstring at *de* havde siddet på en tagterasse med udsigt til Colosseum og fået en herlig frokost. Samtidig havde vi andre været i Sankta Palomba og set en dum og kedelig fabrik - fabrikker er altid kedelig fordi alting sker automatisk så der er ikke rigtig noget at se på.

Heldigvis havde vi en dag for os selv. Jeg husker vi travede og travede og travede, og vi blev meget trætte og ømme. Men vi fik da bl.a. set Piazza Navona, Peterskirken og Fontana di Trevi. Trætte vendte vi hjem til Hotellet og fik en lille forfriskning. Så blev vi enige om at Borgheseparken ikke var så langt væk, så vi begav os derop. I Borgheseparken fik vi en pragtfuld øl, jeg tror grunden til at den var så god, var at vi trængte gevaldigt til den.

Det fascinerede ved Rom er, at man aldrig kan få nok. Man kan blive ved og ved, fordi der hele tiden er noget nyt (gammelt??). Det er ikke IBM, der er skyld i at jeg kender Rom, men jeg er IBM meget taknemmelig for at de har sendt mig derned fem gange.

Der er en anden fordel ved sådan en tur. Kirsten fik et helt anderledes positivt indtryk af IBM. Det var som om hun bagefter meget lettere kunne forstå at jeg til tider forsømte familien, fordi jeg mente at mit arbejde var vigtigt - altså ikke vigtigere end familien, men alligevel vigtigt. Pludselig kunne hun forstå, hvorfor jeg var så glad for mit arbejde og for IBM.

Medens jeg havde travlt med at passe IBM's VM-kunder, havde Jon udviklet en af sine genistreger, den han blev mest berømt for. Det var PIPELINES. Et system som CMS har naturligvis en række kommandoer, mere eller mindre komplicerede, der gør det muligt at gøre det meste af det, man har behov for.

Et system som UNIX har den smarte ting at man kan tage output fra én kommando, og bruge det som input til en anden kommando. Man kan på en måde lade data glide som en strøm, fra kommando til kommando. Det kaldes 'piping' altså noget med rør. Nu har VM/CMS altid været kendt for at man *der* kunne gøre alt, hvad andre systemer kunne, blot lidt mere elegant. Selvfølgelig kunne Jon ikke have siddende på VM at man kunne pipe med UNIX og ikke med CMS, så derfor gik han i gang med at skrive sine PIPELINES, der nok blev noget af det mest elegante, der nogen side er sket med CMS.

PIPELINES virker på den måde at man har nogle device-drivere til at læse data, og efter behandlingen at skrive data, desuden er der en række (rigtig mange) filtre, der kan gøre alt muligt (og måske også umuligt) med disse data. Uden PIPELINES kan man f.eks. have én kommando, der læser data, gør noget ved dem, skriver dem på

disken, hvorefter en anden eller flere kommandoer igen skal læse, behandle og skrive. Med PIPELINES læser man data og behandler dem med ét eller flere filtre, hvorefter de skrives der hvor man skal have dem, det kan være på tape, disk eller printer. Det er helt klart at denne filosofi resulterer i programmer, der benytter færre ressourcer og som kører langt hurtigere, end ved mere konventionel programmering bl.a. Fordi data kun skal læses og skrives én gang.

For at sådan et program kan blive et produkt, man kan sælge til kunder, skal det igennem en validerings-proces, der bl.a. består i at programmet skal installeres på en 'fremmed' maskine - altså en anden, end den, det er skrevet og udviklet på, og af en uvildig person altså nok én der ikke kender programmet for godt. Jon havde fået en god ven i England, i øvrigt en af de bedste VM'ere jeg kender, så han havde bedt ham om at validere PIPELINES. Imidlertid havde han meget travlt, så det trak ud, og Jon var naturligvis utålmodig, han ville gerne have sin 'baby' frem i lyset. Han kom så i tanke om at jeg havde kontor på en anden adresse end IBM's hovedkontor, og jeg havde min egen private VM-maskine. Det er naturligvis ikke *helt* rigtig, men min afdeling havde sin egen lille maskine. Den blev brugt til til uddannelse og demonstrationer, men da jeg var VM-manden på stedet, var det naturligvis mig, der vedligeholdte programmet på maskinen - *altså min maskine!!* Denne maskine var altså helt uafhængig af IBM's standard system. Derfor kunne den bruges til validering. Selvfølgelig ville jeg da gerne gøre Jon den tjeneste. Det var et kvalitetsprodukt - naturligvis. Jeg tror kun jeg fandt ét lille problem. Det var *den* del af programmet Jon ikke havde været i stand til at teste, da det krævede en mindre om-definering i selve VM. Det tog vist Jon ca. 5 minutter at finde og rette fejlen, så det var en ren fornøjelse. IBM har i øvrigt for vane at udbetale 500\$ for ulejligheden. Desværre for mig ventede man så længe med at finde pungen frem, at dollar-kursen i mellemtiden var faldet betydelig.

Uheldigvis for mig, blev PIPELINES ikke standard i CMS, før jeg holdt op med at være VM specialist. Det jeg i tidens løb har udviklet, har altid baseret sig på standard elementer af VM, eller på ting jeg selv har skrevet, jeg har med andre ord ikke villet være afhængig af at en eventuel bruger af noget jeg har lavet, skulle være nødt til at anskaffe yderligere programmel, for at kunne bruge det. Derfor kom jeg ikke rigtig i gang med at benytte PIPELINES, jeg blev aldrig en rigtig 'PIPER'. Senere har jeg dog udviklet en hel del applikationer til CMS, hvor jeg flittigt har anvendt PIPELINES. Mange af disse programmer ville måske slet ikke kunne skrives uden!!!

Som så mange andre begyndte jeg med at bruge PIPELINES til at læse data ind mit i program. Så kunne jeg behandle dem med standard REXX-kode, ganske som jeg var vant til. Naturligvis opdagede jeg efterhånden, at der *lige* var det her filter, som jeg lige så godt kunne bruge, når jeg nu havde data i 'pipen'.

Pludselig en dag, gik det op for mig, at hele min opgave var løst, ved hjælp af PIPELINES. Programmet bestod reelt af én instruktion: en PIPE-kommando, der til gengæld indeholdt en hel del filtre. Programmering på denne måde er *meget*

effektiv, og programmerne er særdeles effektive, og på grund af Jon's store evner som programmør, er der ikke mange problemer i systemet hvis overhovedet nogle, på trods af at det har så utrolig mange muligheder. I følge Jon selv, ville man kunne skrive ca. 80% af CMS's kommandoer i PIPELINE (eller var det kun 70% ??).

Jeg havde en overgang en afdelingschef (altså min chefs chef), der var umådelig kreativ. Han havde den usædvanlige opfattelse af computere, at de var til for at hjælpe folk i almindelighed og ham i særdeleshed. Han stødte ustandselig på ting i sin hverdag, der irriterede ham, og naturligvis måtte jeg høre på disse beklagelser, da jeg var den, der kendte systemet bedst. Jeg forsøger til tider at være venlig - specielt over for mine chefer - så jeg lavede en række småprogrammer, der hjalp ham over de værste af hans problemer. Det kunne han godt lide, så der kom flere og flere af den slag småopgaver. En af disse opgaver blev faktisk ret stor, jeg fik konstrueret et helt lille menu-system, der var så snedig indrettet at brugeren selv kunne udvide systemet med nye menuer. Da det ikke var særlig kompliceret og stort, hed det naturligvis MINIMENU. Jeg synes selv det var ret elegant, og det var skrevet med kun ca. 1000-liniers REXX-kode inklusiv forklaringer på skærmen på dansk og engelsk, så det var egentlig forbløffende enkelt. Alle, der fik demonstreret MINIMENU, syntes om det, men IBM havde et andet langt større menu-system, som man ihærdigt solgte til kunder med stor succes, så min afdeling syntes ikke man ville konkurrere med dette produkt, MINIMENU blev altså ret hurtigt gemt af vejen i en dyb og mørk skuffe. Lidt trist måske, men det havde været vældig sjovt at udvikle. Cheferen viste sin taknemmelighed ved en af de traditionelle middags-invitationer - altså til mig og fru, men senere gjorde han noget, der fik umådelig stor betydning for min seneste tid i IBM, men mere herom senere.

På trods af at jeg kom vældig godt ud af det, med denne afdelingschef, husker jeg vi en gang havde en konflikt. Man havde fundet ud af at vore kunder manglede kvalificerede systemprogrammører, så det blev besluttet at lave en crash-uddannelse, altså noget, der skulle gå stærk og være effektivt. Naturligvis blev jeg rodet ind i VM uddannelsen, jeg skulle lave et kursus om VM internal - altså, om hvordan VM virker. Nu er sådan et kursus ikke noget man klarer, bare ved improvisation, så jeg gik i gang med at forberede mig, jeg ville jo helst give kunderne noget, der var kvalitet. Så kom der en særdeles kritisk kundesituation, der tilsyneladende ikke kunne klares uden min medvirken - der *har* været perioder, hvor jeg har følt at min betydning måske var større end min løn, og det her var vist én af dem. Sommetider følte jeg virkelig at jeg var en ikke helt ubetydelig person. Jeg forklarede altså chefen at jeg ikke kunne holde kurset til den fastsatte tid, hvis jeg også skulle løse hans kundeproblem. Det var *ikke* velset!! Jeg fik en lang forklaring om at jeg fik 80% af min løn for at være fleksibel og kun 20% for at være dygtig. Det har jeg citeret ham for mange gange siden og altid til hans irritation. Jeg forklarede ham at jeg ikke ville sætte min og IBM's professionelle anseelse over styr ved

at holde et dårligt kursus, og at det i øvrigt også ville være synd for kunderne, samtidig med at det altid er meningsløst at gøre noget, hvis man ikke har til hensigt at gøre det ordentligt - altså en helt trivial gang salgssnak. Jeg tror mine argumenter bed på ham. Han var ellers ikke let at klare i en diskussion, han var en *meget* dygtig sælger, men han gav sig og udsatte kurset, som i øvrigt gik vældig godt. Omkring eleverne var der dog den besynderlige ting at alle medarbejderne fra et af de delta-gende firmaer havde opsagt deres stilling i protest mod firmaets personalepolitik, men det forhindrede altså ikke at de kom på kurset. Set fra IBM's synspunkt gjorde det dog ikke noget, for de fik jo uddannelsen, så de kunne fungere som system-programmører enten i deres nuværende firma eller i et andet.

Mit sidste residency kom mærkelig nok til at gå til Böblingen, der ligger i nærheden af Stuttgart. Böblingen blev normalt anset for at være 'VSE-land', men IBM i Böblingen har en fabrik, der bygger 'små' maskiner, og disse maskiner blev forbedret/moderniseret, så de kunne køre den nye 'ESA' arkitektur. Fordi maskinerne var blevet udviklet i Böblingen, havde man så også fået ansvaret for annonceringen af de tilhørende programmer i Böblingen - blandt disse programmer var naturligvis VM. Derfor!!! Det var et enmands residency, og jeg blev uheldigvis indlogeret på et hotel, hvor der ikke boede andre ligesindede, så det blev en temmelig ensom affære. Jeg gik altså glip af en af de største fordele ved sådan en tur, nemlig kontakten til udenlandske kolleger. Selvfølgelig traf jeg da andre, men da vi ikke arbejdede sammen, blev der ikke særlig megen forbindelse. Jeg traf dog en ung pige fra Dallas, hun var vistnok uddannelseschef. Da hun hørte jeg var dansk, fortalte hun at hun havde været på bryllupsrejse til Danmark. Naturligvis spurgte jeg hende hvor hun havde været. Det fik hende til at se helt forkert ud, hun gik så hen til et whiteboard og tegnede ordet "ÆRØ" på tavlen. Et stednavn på tre bogstaver hvoraf hun ikke kendte de to, var simpelthen for meget for hende, men hun havde i øvrigt været meget glad for sit ophold i Danmark.

Som jeg havde forventet, var der ikke mange i Böblingen, der havde særlig meget kendskab til VM, min projektleder kontaktede flere gange sin specialist, for at få hjælp, når han og jeg var uenige. Heldigvis gav specialisten mig ret hver gang, så jeg slap for at kontakte *min* specialist (Jon), men i følge Jon var denne tyske specialist ikke særlig klog, så det var nok heldigt der ikke kom de helt store diskussioner. I løbet af de fire uger, jeg var i Böblingen fik jeg da produceret en række foils, som *måske* blev brugt, så min målsætning blev altså opfyldt.

Den første dag i centeret, blev vi alle inviteret ned på et ydmygt sted, der hedder Bierbrunnen (det betyder vistnok ølbrønden - tysk har aldrig været min stærke side), et prægtigt sted, hvor man kunne få fadøl, som det *bør* serveres (mindst 15 forskellige slags). Jeg kom til at sidde ved siden af og sludre med en fyr fra fabrikken. Da han hørte jeg skulle arbejde med VM, kom han med nogle foruroligende udtalelser om de nye maskiner. Udtalelser, som, hvis de var rigtige, ville gøre at maskinerne *ikke* ville kunne køre VM ordentligt. Det drejede sig om en

maskininstruktion, der *måske* manglede, en instruktion, der tager hånd om virtuelle maskiner, så det var *virkelig* en vigtig nyhed for mig. Instruktionen, der hedder SIE (Start Interpretive Execution) var bygget specielt for at få VM til at køre mere effektivt, når den kørte med gæste operativsystemer, men mærkværdigvis var der *ingen* i centret, der havde hørt om denne instruks, ikke en gang deres to hardware specialister. Endnu en gang gik det op for mig at jeg måske var VM specialist på et lidt højere plan end mange andre. Ved hjælp af Jon fandt jeg nummeret på den manual, der beskrev SIE instruktionen, så jeg bad en af de lokale om at rekvirere den til mig. Da de så SIE beskrevet i en manual for sig selv, begyndte de endelig at tro på mig. Jeg fik lov at låne nogle af de *meget* fortrolige bøger om de nye maskiner for at undersøge sagen nærmere. Sådan nogle bøger er stribe, næsten som Bismarks klumper, og de blev opbevaret i pengeskabe med kombinations lås og måtte kun læses i isolerede kontorer, ikke noget med at tilfældige skulle kunne læse over skulderen. Jeg tror det meste af mit 4-ugers residency var gået før jeg endelig fik bekræftet at der rent faktisk *var* SIE instruktioner på de nye maskiner, så de *kunne* altså køre VM, hvilket gjorde mit arbejde dernede en hel del mere meningsfyldt.

Tyskland har mange smukke middelalder byer, men Böblingen er *ikke* en af dem. Böblingen er nok noget af det mest deprimerende man kan forestille sig. Til mit held havde min projektleder hverken bil eller kørekort (det var noget med hans øjne og sukkersyge), men han kunne vældig godt lide at køre rundt i omegnen, så vi kørte et par ture sammen. Det bevirkede naturligvis at jeg fik set noget, jeg ellers ikke ville have set. Hans lokalkendskab var angivelig meget bedre end mit. Jeg kom blandt andet til den gamle hyggelige universitetsby Tübingen, der ligger kun ca. 30 km syd for Böblingen. Tübingen er meget hyggelig, fyldt med smukke gamle bindingsværkshuse af den slags som gør mange tyske byer så maleriske. I Tübingen gik det pludselig op for mig, at der var rigtig langt til havet, og så fik jeg hjemve. Det var vist første gang det rigtig gik op for mig, at det betyder utrolig meget for mig at bo nær havet. På en anden tur, fik vi set det sjoveste lille slot 'Sloss Lichtenstein'. Det er ikke meget større end en gennemsnitlig villa og det er bygget på en klippetop, der rager op gennem gulvet i en af salene. Man skal passere en vindebro for at komme til dette lille eventyr/torneroseslot.

En anden weekend kørte jeg alene gennem Schwartzwald ad Schwartzwald-hochstrasse. En dejlig tur, som jeg dog ikke fik det fulde udbytte af på grund af ret dårligt vejr, men det var ikke første gang jeg havde problemer med vejret.

Jeg tror nok, jeg fik lavet et pænt stykke arbejde i Böblingen, de skrev i hvertfald at de gerne ville se mig igen, hvis der kom en passende lejlighed. Jeg har altid været rimelig grundig, når jeg arbejdede med VM, sikkert fordi jeg synes så godt om produktet, så naturligvis gjorde jeg mig også umage i Böblingen.

Senere har jeg hørt at Böblingen i højere grad er blevet et center for VM-viden - endnu mere end Poughkeepsie - måske var *jeg* med til at starte denne udvikling???

Det var begyndt at gå mindre godt for IBM, så vi fik alle besked på at spare. En af besparelserne ville man opnå ved at stoppe med at købe planlægningskalendere til medarbejderne. Jeg havde min TimeSystem, som jeg var meget glad for, så det passede mig rigtig dårligt. Hvad gør man så??? Man laver naturligvis sig egen kalender i et format, der passer til TimeSystem's ringbind. Det tog vel et par dage plus noget finjustering. Den seneste rettelse blev først lavet et par år efter jeg var blevet pensioneret, det var noget med ugenummeret. Jeg måtte have fat i Mayland for at få at vide at uge ét er den første uge i året, der har mindst fire dage i sig - mit leksikon påstår noget andet, så derfor havde jeg programmeret forkert. Min kalender blev ret populær, der var ganske mange IBM'ere, der pludselig var interesseret i at få læse adgang til den disk, hvor kalenderen lå. Der var oven i købet en chef, der begyndte at tale om at sælge den. Det fik jeg nu hurtigt talt ham fra. Jeg er ikke klar over om der stadig er en afgift til universitetet og sådan noget, og det vidste han heller ikke og måske lignede den også TimeSystem lovlig meget? Til orientering har jeg senere - meget senere, fået konverteret kalenderen, så den nu kan køres ved hjælp af REXX og WordPro. Mærkelig nok er PC løsningen mere elegant end VM løsningen, bl.a. fordi PC kalenderen har mulighed for private mærkedage og fødselsdage. Når kalenderen skal bruges privat, kommer der jo helt naturlig en række nye krav også til formatet, og så lykkedes det endelig for mig at finde ud af hvornår det er påske, så jeg kunne få de variable helligdage på plads. De der variable helligdage er ret indviklede, men 'Encyclopædia Britannica' (på internettet) havde en løsning der let lod sig omsætte til REXX.

En dag blev jeg ringet op af før omtalte kreative afdelingschef. Han var nu i en helt anden afdeling. Han syntes at CMS efterhånden var ved at blive lidt gammeldags og at jeg skulle begynde at interessere mig for noget mere moderne. Han havde en PC til overs og den ville han gerne have mig til at lege med - ingen betingelser eller bagtanker. Selvfølgelig ville jeg da gerne prøve at lege med en PC, så jeg hentede et vældigt skrummel, og fik udleveret et par DOS manualer. Jeg gik i gang, ikke med at lave noget, men med at finde ud af hvad én med min baggrund, kunne bruge sådan en æske til. Der gik nu ikke lang tid, før jeg fik besked på at aflevere maskinen og hente en anden og større, én der kunne køre OS/2. *Nu* var der vistnok en lille bagtanke, men det dukkede først op noget senere. OS/2 havde eller fik ret hurtig, mulighed for at køre REXX, og så var jeg *næsten* på hjemmebane. Det var vældig sjovt, og naturligvis fortalte jeg min 'sponsor' om mine opdagelser og fremskridt. Så kom det!! På VM er der en editor, der hedder LEXX. Den er vistnok specielt skrevet til ordbøger, men forfatteren havde skrevet en LEXX makro, der kunne lave GML om til SGML. GML er det tekstbehandlingssystem SCRIPT havde udviklet sig til, og SGML var noget tilsvarende, der dog kunne bruges på en

PC. Det gav med andre ord mulighed for at konvertere VM tekst til PC tekst, og min sponsor var meget interesseret i at flytte arbejde fra de store værtsmaskiner til PC. LEXX fandtes i en PC udgave under navnet LPEX, og jeg fik nu til opgave at prøve at ændre VM-makroen - altså 'GML til SMGL' - til noget, der kunne køres på en PC. Det er nok ikke helt fair at udtrykke at jeg fik det til opgave, men det blev mere end antydnet at det ville glæde . . . hvis det kunne lykkes. Det lykkedes. Min PC var dog alt for lille til at køre OS/2, så det gik *meget meget* langsomt, men belønningen kom hurtigt. Naturligvis var vi henne om hjørnet for at fejre - øl og sådan, men vigtigere var det nok at jeg fik en mere moderne og større PC, der var særdeles velegnet til OS/2. Jeg blev yderligere forsynet med en laserprinter og et modem, så jeg kunne kontakte mit kære VM-system hjemme fra Kokkedal. Modem'et var ganske vist kun på 2400 bps, så det var så langsomt at det *ikke* var praktisk muligt at arbejde hjemmefra, men jeg kunne da klare den elektroniske post uden at skulle ud af døren. Joh, det var sandelig noget, der pyntede på min hverdag og noget jeg havde drømt om længe.

PC'erne var blevet større, og de var efterhånden blevet ret populære, og VM og de store maskiner var ved at glide i baggrunden. Måske var mine kunder også blevet for dygtige så de selv kunne løse de fleste af deres problemer. Jeg har hele tiden ment at det var en vigtig del af mit job (måske den vigtigste), at uddanne kunderne, så de bedre kunne klare sig, måske var det lykkedes for mig - næsten for godt??.

Pludselig en dag fik jeg at vide at jeg ikke mere var VM specialist. Det var noget af et chok. Jeg havde hele min aktive tid i IBM været specialist inden for et ret snævert område, og nu skulle jeg være systemkonsulent med ansvar for et par kunder, det man kalder account-SE. Sådan en fyr skal helst vide lidt om alting i stedet for alting om meget lidt, han har jo til opgave at snakke med kunderne om *alt* det kunderne kan finde på at tale om, for ikke at tale om at der også var visse administrative opgaver - og administration har *aldrig* været mig. Yderligere skal han naturligvis hjælpe sælgeren med alle mulige tekniske spørgsmål, altså ikke bare VM spørgsmål - men alt muligt. Det var *slet* ikke mig. Jeg gjorde mit bedste, men jeg kunne over hovedet ikke trives i den rolle, så jeg vandrede den tunge vej ind til min chef for at beklage mig.

En aften ringede han så til mig og fortalte at jeg skulle smutte ud og snakke med chefen for CMSC (Competitive Marketing Support Center), de manglede en mand som måske godt kunne være mig. CMSC blev så den afdeling i IBM, hvor jeg tilbragte mine sidste aktive år.

This page intentionally left blank.

OS/2 og LAN

Jeg var nu havnet i CMSC, det er forkortelse af Competitive Marketig Support Center, den afdeling i IBM, der holder øje med konkurrenterne. CMSC var ikke en afdeling jeg kendte meget til, jeg havde en god kollega, der snakkede meget om at han godt kunne tænke sig at arbejde der, men jeg fik vist aldrig at vide hvorfor. CMSC havde den ulempe at de var placeret i EJBY, altså meget længere fra Kokkedal end Lundtofte, og den længere vej var mindst lige så populær, som den jeg var vant til, der var med andre ord en forfærdelig trafik - hele vejen. Det var som om jeg skulle have længere til arbejde hver gang jeg skiftede job, til gengæld havde de den store fordel at alle havde eget kontor - altså gode arbejdsforhold.

CMSC havde en helt anden holdning til EDB, end den, jeg var vant til. Man havde lyttet til nogle anbefalinger, der gik ud på at man skulle spare på anvendelsen af værtssystemerne, og bruge PC'ere så meget som muligt. Det gjorde man ret avanceret. Man havde tidligt anskaffet PC'ere til alle medarbejdere, og man satsede virkelig på at anvende disse PC'ere så man kunne spare på de dyre ressourcer fra værtssystemerne. Min opgave blev at give intern støtte til CMSC'erne og administrere og vedligeholde CMSC's computernetværk, LAN (Local Area Network). Naturligvis kørte man OS/2, som jeg havde erfaring med, fordi jeg, som tidligere fortalt, havde lånt en PC, og netop brugt den til at få erfaring med OS/2. Den lånte PC var for længst afleveret, min sponsor havde fået nyt job i en anden afdeling, så jeg måtte læsse bilen og aflevere alt maskineriet. Det blev et *stort* savn. Når man først har vænnet sig til at have sådan et skrummel stående, er det slet ikke sjovt pludselig at skulle undvære den, men heldigvis kom IBM samtidig med et forholdsvis favorabelt tilbud på hjemmeterminaler. Det var noget med at skattevæsenet skulle hjælpe med at betale dem, så jeg havde anskaffet den mest beskedne, jeg kunne få.

Det var et stort held for mig. Godt nok kendte jeg OS/2, men ikke på det niveau jeg nu havde brug for, og jeg kendte slet ikke de programmer CMSC benyttede under OS/2. Og så var der LAN. Jeg vidste, hvad ordene betød, jeg havde en vag fornemmelse af, hvad det var, og det var så det. Jeg fik dog et par små kurser, og det hjalp lidt, men jeg må erkende at jeg var glad for at have et par særdeles flinke kolleger der hjalp mig, når det kneb. LAN ekspert blev jeg aldrig, jeg havde virkelig behov for hjælp, nu havde jeg i ganske mange år været den, der vidste (næsten) det hele, så det var en helt anderledes og lidt utryg situation, jeg var havnet i.

På det lav, hvor jeg fik min hjemmeterminal, begyndte min søn at interessere sig for PC'ere, og da jeg kom hjem med en pakke med OS/2, skulle han naturligvis prøve. Det gik *slet* ikke. Den unge mand havde temperament, så jeg forsøgte at dysse ham lidt ned ved at få ham til at læse i den bog, der blev leveret med systemet. Det indrømmede han var en god og rimelig idé, så han tøffede ind til sig selv med bogen under armen. Omkring et kvarter senere, dukkede han op og sagde: "Det går ikke Far. Jeg kan ikke forstå det. Det er på Dansk". Jeg har tit følt lidt uhygge ved denne lille episode. Nogen gør sig vældig umage med at oversætte computer litteraturen til f.eks. dansk, men gør det på en måde, så resultatet bliver

helt uforståelig for fagfolk. Fejlen er at man opfinder nye danske udtryk, i stedet for at benyttet de engelske, der som regel let kan fordanskes. Det er for dumt at man er nødt til at lære en dobbelt terminologi, bare fordi man er så uheldig at være født i et ikke engelsk-talende land.

Jeg efterlyser ikke snobberi for Engelsk, jeg synes blot det er mest praktisk med så få specielle termer som muligt. F.eks. blev en printer ret hurtigt fordansket til 'linieskriver', jo det lyder måske lidt mere dansk, men en laser-printer skriver ikke linie for linie, og så er ordet 'lineskriver' vel egentlig noget vrøvl, og ordet printer kan vel lige så godt være et dansk ord som lineskriver??

Jeg kom senere til et 25-års jubilæum hos en kollega - det var ham der oprindeligt havde fået mig 'løkket' ind i VM-verdenen. Han var i mellemtiden havnet i en afdeling, der beskæftigede sig med maskinel oversættelse og sådan noget. Naturligvis havde han kolleger fra sin egen afdeling med, og da jeg fortalte om episoden med min søn, blev det anledning til en ret hidsig diskussion mellem mig og et par unge damer. Der var en yngre medarbejder fra hans afdeling, der følte sig lidt trængt af mine argumenter, så hun søgte hjælp hos en mere erfaren, der efter at have lyttet til mig, gav mig fuldstændig ret.

Jeg fik rigtig mange problemer i mit nye job, det er f.eks. ikke morsomt at stå alene med en LAN-server, der absolut ikke vil køre, og som er utilbøjelig til at fortælle, hvad der er galt, når man ved at der sidder en hel afdeling, der er afhængig af informationer på LAN'en. Et par af dem stod naturligvis interesserede og fulgte mine anstrengelser, hvad der *slut* ikke hjalp. Jeg fandt efterhånden ud af at der vistnok var for lidt plads på serverens diske, men hvorfor??? Det viste sig at der var et bibliotek med maser af ikoner (det er de der små dekorative billeder, der er på skærmen, og som repræsenterer et program eller et bibliotek). Sådan en ikon fylder fra små 1000 bytes og op efter. Altså ikke særlig meget, men hvis der er tilstrækkelig mange tusinde, kan det godt blive til en anseelig mængde. Da jeg slettede dem, begyndte serveren at køre igen, og jeg kunne tørre sveden af panden, men jeg husker tydeligt at det tog ganske lang tid at slippe af med alle de der ikoner, så der var vist adskillige tusinde.

En af de store forskelle på den verden med værtsmaskiner, som jeg kom fra og PC-verdenen, som jeg nu skulle til at leve med, var forskellen på kvaliteten af dokumentation. VM-bøgerne var af meget høj kvalitet - det var jo VM, og forfatterne var forpligtiget til at rette at alle fejl meget hurtigt. Hvis en kunde beklagede sig over noget i en VM bog, skulle han have svar omgående, så vidt jeg husker skulle der reageres på problemet inden 24 timer. PC bøgerne var der imod ganske anderledes primitive. Det er egentlig meget mærkeligt, da de normalt er skrevet for at kunne bruges af ikke professionelle. Jeg har mange gange på skærmen set en meddelelse: "Der er noget galt. Tryk på HJÆLP knappen, for at få yderligere information". Det gør man så, og får endnu en gang at vide at der *er* noget galt!!

Problemet er måske at PC programmerne er meget billige og man derfor ikke synes at man kan/vil ofre penge på en ordentlig dokumentation, men jeg synes jeg har haft mange store (og overflødige) problemer netop på grund af alt for dårlig dokumentation. Det gælder både bøgerne og de informationer, der er placeret som en del af programmerne i form af hjælpe information.

Chefen for CMSC havde anbefalet sine medarbejdere at anskaffe en PC på IBM's tilbud om hjemmeterminaler, og han havde lovet at hvis de gjorde det, skulle de få mulighed for at arbejde hjemmefra. Det var meget attraktivt for CMSC's medarbejdere, da de havde stor rejseaktivitet og derfor gerne ville se familien så meget som muligt. Adgang til IBM's værtssystemer hjemmefra var ikke noget nyt, men CMSC var jo en afdeling, der brugte PC og LAN, og adgang til *både* værtssystemerne og LAN, var bestemt noget nyt. Min forgænger i jobbet og en særdeles kommunikations-kyndig israeler, havde fundet en mystisk æske, der kaldtes en chatterbox. Chatterboxen er en kasse, der kan indeholde en række kort, der alle indeholder en PC - naturligvis uden harddisk. Til hvert PC-kort var der et tokenring-kort, der gjorde det muligt for disse små PC'ere at få forbindelse med en LAN. De små PC'ere blev startet ved hjælp af en teknik, der kaldes remote-ipl. Kassen virkede på den måde at man ringede til den fra sin PC. Når man havde forbindelse, skulle man præsentere sig, hvorefter forbindelsen blev afbrudt, og chatterboksen ringede tilbage. Det var først og fremmest af sikkerhedshensyn, men det havde også for brugeren den fordel at det var IBM, der kom til at betale telefonregningen. Når man så havde forbindelse, skulle man logge sig på CMSC's LAN, og derefter kunne man starte en teminal session, så man også kunne få adgang til værts systemerne - altså f.eks. mit kære VM. Chatterboxen har tjent CMSC tro indtil et godt stykke tid efter jeg blev pensioneret. Der er naturligvis i mellemtiden kommet langt bedre, men også mere ressource-krævende metoder til at få systemadgang hjemmefra, men fordi mange hjemmeterminaler er ret små, har chatterboxen altså været i brug indtil den for nylig pludselig afgik ved døden. Manden, der i sin tid, installerede den, har for længst forladt CMSC, IBM og vistnok også Danmark, og han er nok den eneste, der har mulighed for at bringe kassen til live igen. Jeg var den, der kørte på chatterboxen, da den døde, men jeg er nu temmelig sikker på jeg ikke gjorde noget forkert, og sådan en æske må naturligvis ikke kunne få problemer på grund af noget en bruger kan finde på at gøre.

Sikkerhed og EDB er noget underligt noget. Jeg har altid ment at IBM's største sikkerhedsproblem er, at de mennesker, der laver reglerne ikke kender noget som helst til elektronisk databehandling!!! Vi fik nogle nye regler. Alle PC'erne skulle være aflåst formodentlig for at forhindre at harddiskene blev stjålet. Det er ikke noget stort problem, hvis man har nøglerne, og hvis man kan se, hvilke nøgler, der passe til hvilke PC'ere, men *det* var netop problemet. Vi havde en masse maskiner, og en skuffe fuld af nøgler. Det er meget frustrerende at skulle prøve samtlige

nøgler - for at finde ud af at den allersidste, er den rigtige. Det lykkedes faktisk for mig adskillige gange hvorimod jeg ikke tror jeg en eneste gang var heldig og fandt den rigtige nøgle ved første forsøg. Efterhånden fandt jeg de fleste nøgler og - ikke mindst vigtigt - jeg fik sat små navneskilte på dem, så de fremover var lidt lettere at identificere. Jeg kan stadig ikke se en PC nøgle uden at få kuldegysninger, det var et *rigtig* stort og enormt kedelig arbejde måske også fordi nogle af maskinerne var anbragt lettere utilgængelige.

Da dette projekt omsider var overstået, og jeg troede at jeg skulle til at slappe af, var der én, der fandt ud af at IBM skulle have lidt styr på maskinparken. Måske ikke så mærkeligt, når man tænker på, hvor mange maskiner et firma som IBM har spredt rundt omkring. Nu måtte jeg rundt og finde serienumre på alle vore maskiner. Heldigvis har jeg en lille lommelygte, og den fik jeg sandelig god brug for. Serienumrene er anbragt de mest umulige steder på maskinerne. Det er næsten som om man skammer sig over disse numre. Når man så tænker på, hvor mange mærkelige steder folk kan finde på at stille deres PC, for at den skal stå mindst mulig i vejen!! Jeg husker jeg kravlede rundt under borde og andre mærkelige steder, for til sidst at finde ud af at de maskiner, vi havde slet ikke svarede til de maskiner, man troede vi havde. Nogen ville så gerne vide hvorfor???? Jeg har vist før nævnt at administration ikke er min stærke side og ej heller har min store interesse.

CMSC havde VM programmer kørende, da jeg kom. En dag kom en kollega og bad mig ændre i en procedure han havde skrevet. Der skulle indlægges et par ekstra funktioner, han mente - og med rette - at det ville være lettere for mig at skrive den pågældende ændring. Jeg fik yderligere oplyst at programmet normalt kørte i 45 til 50 minutter. Jeg fandt hurtigt ud af hvordan programmet skulle udbygges, men da jeg derefter så, hvordan det var kodet, skrev jeg det lidt om, således at det fremover i stedet kørte på 5 - 6 minutter. En ikke helt ringe forbedring. Moralen er at det ikke er helt ligegyldigt, hvordan man skriver sine programmer - eller måske: det er bedre at benytte en programmør med lidt erfaring.

Jeg husker også en anden situation, hvor CMSC fik glæde af at have en VM specialist i 'flokken'. Der kom en ny mand, og han skulle naturligvis have adgang til nogle CMSC specifikke diske. Sådan en adgang gives centralt, og kun ejeren af disse data kan anmode om det. Ejeren var min chef, der var på ferie. Vi måtte altså vente. Det fandt jeg usmart, så jeg opfandt en maskine, der skulle eje disse diske. Det var en speciel maskine, som andre udvalgte brugere kunne få adgang til ved hjælp af deres eget password, en af VM's glimrende faciliteter. Vi var altså nogle stykker, der kunne køre på denne maskine, og derfra sende anmodning om disk adgang. Dette betød at anmodningen altid kom fra ejeren, og så var det problem løst - uden datasikkerheden blev brudt!!!! Jeg har altid været meget motiveret for at arbejde for datasikkerhed og altså endnu en elegant VM/CMS løsning.

Hvordan trivedes sådan en gammel VM'er som mig så sammen med OS/2??? Det gik faktisk rigtig godt. Jeg har de sidste mange år udviklet, hvad jeg skulle bruge, ved hjælp af programmeringssproget REXX. Heldigvis kan OS/2 også køre REXX, og slet ikke dårligt. Ikke helt den samme, som VM men dog REXX. Desværre mangler OS/2 de navnkundige PIPELINES. Jeg har tigget og bedt Jon, men han vil gerne have penge for at lave det, måske ikke så mærkeligt, *han* kan vist også godt lide at få noget at spise ind i mellem (han er i hvert fald meget dygtig til at lave mad). Men *jeg* har jeg ikke råd til at betale for pipelines til OS/2, så det må jeg altså undvære. Et vigtigt værktøj til programudvikling er en tekst-editor, og som tidligere nævnt, er XEDIT noget af det mest fremragende, der er fremstillet af den slags. Heldigvis findes der en PC udgave af denne editor, kaldet KEDIT (man skal ikke vride ret meget på benene i et X, for at lave det om til et K). Min PC sponsor gav mig tidligt en licens til KEDIT - det er ikke et IBM produkt, så det koster 'rigtige' penge, men han syntes at jeg havde fortjent at få et godt værktøj. Måske var det også belønning for noget af det, jeg havde lavet til ham.

Med REXX og KEDIT, kan man lave næsten det samme som man kan under VM/CMS. Det er nok ikke helt så elegant, men til gengæld ser det meget flottere ud, fordi PC skærmene er langt bedre end dem, man brugte til værtssystemer som f.eks. VM. Nu om dage benytter næsten alle PC'ere til adgang til værtssystemer, men værtssystemerne er jo ikke udviklet til den type skærme så man kan ikke udnytte alle skærmens muligheder fra værtssystemerne.

Til REXX findes en slags performance måleværktøj, REXXCPS. Der måler hvor mange typiske REXX instruktioner, man kan udføre pr. sekund. På IBM's VM system, viser REXXCPS typisk ca. 35000, målt på en søndag, hvor maskinerne ikke har meget at lave. Min beskedne PC kan klare ca. 4500, så det er slet ikke så galt at køre REXX på en PC.

Så bundlinien viser at jeg har det *helt* fint med OS/2.

Arbejdet i CMSC var rimeligt afvekslende. End-users, de hedder vistnok slutbrugere på 'dansk', er ind imellem meget ubehjælpssomme, "*Det virker ikke*", var noget jeg hørte ustandseligt. Det pudsige er, at jeg i de fleste tilfælde kunne løse problemet ved at slukke og tænde for maskinen. Hvis man 'roder' tilstrækkeligt meget rundt i et system som OS/2, uden på noget tidspunkt at tænke på at lidt 'oprydning' kunne være på sin plads, som f.eks. at lukke programmer, man ikke mere bruger, kan man risikere at en eller anden ressource - formentlig hukommelse - bliver opbrugt. Der findes f.eks. programmer, der fejlagtig undlader at frigive al den hukommelse, de har brugt. Hvis disse programmer kører mange gange, får man mindre og mindre hukommelse til rådighed. Når man ikke aner, hvad der er sket på sådan et system, er den enkleste form for oprydning at slukke og tænde maskinen, en glimrende løsning i de fleste tilfælde, fordi det var yderst sjældent der var vigtige data, der gik tabt. Det kaldes i fagsproget 'Big White

Switch' på trods af at kontakten ofte er rød eller har en anden farve. Det er nok lidt som det der med 'Den Sorte Kasse', man altid taler om ved flyveulykker. Disse kasser er vist nok orangefarvet, for at gøre det lettere at finde dem.

Selv om jeg havde fået et komplet nyt job, havde jeg naturligvis ikke glemt, hvad jeg kunne i forvejen. CMSC havde til opgave at samle informationer om IBM's konkurrenter, for at kunne hjælpe kolleger i konkurrencesituationer og sådanne informationer skulle naturligvis dokumenteres. Det gik imidlertid hurtigt op for sekretariatet at jeg vidste noget om SCRIPT, så jeg blev hyppigt indraget i den slags arbejde. Som tidligere nævnt er jeg meget interesseret i tekstbehandling, så det passede mig rigtig udmærket.

Så var der naturligvis gode gamle VM. VM har jo i mange år været hjørnестenen i IBM's interne databehandling og kommunikation, og mange vigtige applikationer ligger under VM. F.eks. er der en hel del databaser og en række af de såkaldte TOOLS-maskiner. Det er VM virtuelle maskiner, der indeholder programmer og data, som alle kan få adgang til. For nogle TOOLS-maskiner, er der dog restriktioner, det gælder bl.a. for CMSC's. De enkelte lande skulle betale for CMSC's services, og det var naturligvis kun de betalende, der havde adgang til disse data. Derfor havde CMSC også brug for VM ekspertise, og derfor blev jeg helt naturligt brugt til en hel del af det, jeg allerhelst ville arbejde med, så det er ikke få VM applikationer jeg fik lavet til CMSC, medens jeg var der. Den slags kan godt gå hen og blive en fordel, men mere herom senere.

Vi fik en morsom opgave, det var noget med at når CMSC's programmanagers havde løst en opgave for en 'kunde', skulle de rapportere tidsforbrug og sådan noget. Det kneb til tider at få disse rapporteringer igennem, fordi det var noget man gjorde 'senere'. En leder fik så den strålende idé at det måtte være muligt at lave rapporteringen medens man skrev/sendte løsningen på problemet. Det blev en rigtig 'Preben' opgave. Jeg måtte hjem for at få ro, fordi der hele tiden kom nye idéer, og det er som bekendt meget svært at skyde efter et bevægeligt mål, men jeg fik lavet et meget elegant program, der gjorde nøjagtigt, hvad man havde brug for, det samlede de nødvendige oplysninger tilføjede nogle automatisk og sendte det hele videre. En af de ting der skulle tilføjes til disse data var ugenummer. Jeg lavede programmet, så det udregnede ugenummer, som det var beskrevet i mit leksikon. Et par år senere blev jeg ringet op, ugenummeret var forkert! Så var der kun én ting at gøre, jeg ringede til Mayland fordi et kalenderfirma måtte kunne give mig den rette sammenhæng. Naturligvis fik jeg den rigtige forklaring, nemlig at uge nummer ét er den første uge, der indeholder en torsdag (eller som det blev sagt - mindst fire dage), så nu er mit program altså korrekt hvad ugenummer angår - om der er flere fejl?? Ja sikkert, men de er endnu ikke blevet fundet.

Ophavsmanden til projektet prøvede energisk at skaffe mig en middags-belønning, men chefen forstod det vistnok ikke rigtigt, så den eneste belønning, jeg fik, var mine kollegers glæde ved den store lettelse, jeg havde skaffet dem.

Da vi nu havde fået etableret denne service maskine, fandt vi ud af at den kunne bruges til en hel masse andet. Den fik til opgave at sende data til diverse databaser (rundt om på kloden), og jeg fik selv den strålende idé at den kunne bruges til at udskrive store filer. Jon (naturligvis var det ham), havde lavet et genialt udskrivningsprogram, som jeg installerede på service maskinen. Så skrev jeg nogle procedurer, så den var let at sende sine udskrifter til service maskinen i stedet for at skrive dem selv. Fordelen var bl.a. at udskriften fra service maskinen forgik centralt på en printer, der var meget større og hurtigere end den vi havde i afdelingen. Alt i alt blev det et særdeles nyttigt projekt som gjorde utrolig stor gavn. Så det var virkeligt en fornøjelse at have været med til at lave det - måske også fordi det endnu en gang beviste hvor elegante løsninger man kunne lave med VM/CMS, og så glæden ved at jeg kunne bruge mit speciale endnu en gang.

VM - jeg bliver ved at vende tilbage til VM - er et system, der kan køre på en lang række værts systemer fra de helt små op til de største med mange processorer. Men for at gøre VM endnu mere tilgængelig, var der udviklet et PC-kort, der indeholdt en /370-processor og 8 MB hukommelse, således at den kunne køre VM/SP (altså den lidt ældre system /370 udgave af VM). Det var naturligvis noget, der interesserede Jon, så under et besøg på det pågældende laboratorium, havde han udviklet et filter til pipelines, så man fra CMS kunne læse og skrive OS/2 dataset. Det havde man været så taknemmelig for, at man forærede ham et P/370 kort. Det var dog lettere defekt, men det kørte udmærket. Jon havde ingen PC, men jeg havde en stor, kraftig og rummelig én, så vi stoppede kortet ind i 'min' PC. Nu kræver VM en hel del diskplads, men min afdelingschef, var meget interesseret i projektet, så han bevilligede os en passende harddisk. Det gik utroligt smertefrit at installere VM på en PC. Og det *kørte*. Det var en *pragtfuld* fornemmelse igen at have sit eget helt private VM-system. Det var næsten som da jeg i sin tid havde en test-maskine, da jeg var i salgsafdelingen. Min adgang til VM hjemmefra, var i mellemtiden blevet moderniseret, den 'gamle' chatterbox var nu blevet erstattet af '*Lan Distance*', et produkt, der gør at man hjemmefra kan kontakte systemer, så det fungerer fuldkommen som om, man sidder på en lokal LAN-opkoblet terminal. D.v.s. jeg kunne få adgang til de samme data hjemmefra, som fra kontoret. Det gjaldt også mit eget VM-system, og da Jon, som fortalt, havde udviklet et pipeline-filter, der kunne få adgang til OS/2-diske, kunne jeg altså sidde hjemme i Kokkedal og få adgang til harddisken på min PC på kontoret - den var *ikke* en server, så jeg synes det var ret sofistikeret. Jeg beholdt VM-kortet, så længe jeg blev i IBM, men mit håb om at CMSC selv ville investeret penge og energi i et privat VM-system, blev desværre

aldrig til noget, det ville ellers kunne have udskudt mit pensionerings tidspunkt med en hel del år.

IBM var ikke mere så velhavende, som det havde været, så en dag blev jeg kaldt ind til min chef, og fik et tilbud om tidlig pensionering. Det var nok ikke lige det, jeg havde gået og håbet på, men da mit arbejde ikke mere var det, jeg holder mest af altså VM, og jeg af naturen er temmelig mageligt indstillet, var det måske ikke så dårlig en idé. Der blev altså fastsat en dato for sidste arbejdsdag, jeg skulle finde navnene på nogle gode kolleger, som jeg kunne tænke mig at se til min afskedsfrokost, jeg skulle ønske mig nogle gaver og jeg fik da også lovning på en afskeds-godtgørelse.

CMSC's ledelse var nu begyndt at spekulere lidt over tingene, der var jo alle de der VM-ting, jeg havde udviklet, så var der alle de der VM-ting man gerne ville have udviklet og endelig var der måske også noget REXX til OS/2, så man kontaktede personaleafdelingen for at høre om mulighederne for stadig at benytte mine fattige evner. Det kunne der naturligvis *ikke* blive tale om, når man *havde* fået en fratrædelsesgodtgørelse, så var det ud! Der var vist også noget med skattevæsenet, der jo har det med at blande sig på en ubehagelig måde.

Afdelingschefen fik så den strålende idé at jeg kunne blive ansat, på timebasis, i et firma, der var *helt* uafhængig af IBM, f.eks. et lille software firma. CMSC kunne så rekvirere konsulentbistand hos dette firma, og man kunne betinge sig at den pågældende konsulent blev mig. Jeg kendte heldigvis sådan et software firma ganske udmærket. Jeg havde arbejdet sammen med dem i min VM-tid, og direktøren og en af medejerne var tidligere kolleger, og de var helt med på idéen, der var nok også lidt penge for dem. Deres VM-mand var ydermere lige rejst, han var i øvrigt også en tidligere kollega, men de havde altså selv behov for én fyr med lidt VM-kundskab.

Jeg fik altså en aftale med dem en månedstid før jeg sluttede i IBM, og jeg begyndte så småt at glæde mig til ikke mere at skulle tidligt op hver morgen.

Min sidste dag nærmede sig hurtigt - alt for hurtigt syntes jeg. Jeg fik pillet VM-kortet ud af min PC, og afleveret det til Jon, jeg fik kopieret nogle ting som jeg helt bestemt *ikke* kunne undvære, og jeg fik fundet på et par passende gaver, som mine kolleger kunne give mig, fordi de var så glade for at slippe af med mig (eller hvorfor de nu gør sådan noget) jo jeg ved da godt de gerne vil trøste lidt i afskedens svære stund.. Jeg fik blandt andet en Armstrong/Fitzgerald-plade, som Kirsten og jeg længe havde ønsket os.

Den sidste dag blev, som man kan forvente. Der var morgenkaffe i afdelingen, med en masse pæne ord. Da min chef holdt sin lille traditionelle afskedstale, husker jeg hun sluttede med at fortælle mine kolleger, at jeg stadig ville komme - om end i mindre omfang end før. Hendes tale sluttede: "*Vi ses igen på mandag*". Jeg måtte korrigere hende, den pågældende mandag kunne jeg ikke, så det blev først om

tirsdagen. Dagen sluttede med en herlig frokost - IBM har det med at give gode frokoster. Alle fortalte mig hvilken herlig og dygtig fyr jeg havde været gennem alle årene i IBM. Hvorfor mon de så havde bedt mig om at gå??? Kirsten var naturligvis med.

Bagefter

Så var jeg altså blevet pensionist. Nu skulle jeg rigtig slappe af - og nyde tilværelsen. Der var jo lige den der aftale med IBM. De ville gerne se mig to dage om ugen. Det var dog rigtig hyggeligt. De så alle glade ud, når jeg dukkede op, og der var masser at lave. Der var dels vedligeholdelse af det, jeg havde udviklet, men der var også en masse nyt, man gerne ville have udviklet. Jeg gik altså i gang med krum hals.

Under VM har man et system med nogle databaser, der styres af såkaldte Tools-maskiner. Disse databaser kan benyttes over hele kloden af IBM'ere. CMSC havde naturligvis også sådan en tools-maskine med en række databaser, der flittigt blev benyttet af CMSC's kunder, altså andre IBM'ere fra det meste af kloden. Et land fandt en dag ud af at de fik for lidt for de penge, de betalte for CMSC - det varede dog ikke længe - så chefen for CMSC fandt ud af at det skulle være muligt at lukke adgangen til sådan en database individuelt. Ups, det krævede et indgående studium af sikkerheds systemet og dets muligheder i en tools-maskine, men jeg fandt en løsning, der dog ikke var helt simpel. Jeg fik også programmeret og installeret den. Det var under *det* arbejde, jeg rigtig fandt ud af, hvor anvendelig pipelines er!!! Jeg var ret stolt af min løsning, men jeg tror nu aldrig den blev brugt. Der skete noget 'politisk' som gjorde at man ikke mere havde brug for at udelukke brugere fra enkelte lande, og det var så det. Blev jeg så sur?? Næh, hvorfor skulle jeg det. Jeg havde haft det sjovt med at lave systemet, og jeg fik da også penge for det, så "No hard feeling". Jeg blev vel også lidt klogere/dygtigere selv om det måske er mindre interessant, når man nu er tæt ved at holde op.

Da jeg havde arbejdet for IBM et par måneder, blev jeg ringet op af softwarehuset jeg arbejdede for. De havde en VM-kunde, som de gerne ville have mig til at hjælpe med at passe. De ville altså gerne se mig i Gladsaxe en halv dag om ugen. I Gladsaxe programmerede jeg en masse forskellige hjælperutiner i REXX, altså det, jeg var rigtig god til. En af de større ting var et backup system. Problemet med backup er dels af få lavet en brugbar kopi af data, men hvad der måske er ligeså vigtigt er at kunne finde disse data, hvis der bliver brug for dem. Jeg fik lavet et forment system med indbygget bogholderi, det blev testet meget grundig og det kunne alt hvad det skulle kunne. Jeg var netop blevet færdig med det, da man skiftede system, så det blev endnu en af de mange 'smarte' ting jeg i tidens løb har lavet, men som aldrig blev brugt.

Jeg har aldrig tjent så mange penge, som det år, jeg blev pensioneret. Desværre lagde skattevæsnet mærke til det, så jeg fik en betragtelig ekstraregning året efter (heldigvis havde vi ikke brugt alle pengene, så det blev ikke det helt store problem), men det gør altså lidt ondt at sende så mange penge retur til skattefar..

Der gik så et års tids på denne måde. En halv dag i Gladsaxe og to dage på IBM - næsten hver uge, men så blev CMSC reorganiseret, så der ikke var så mange penge til 'løsarbejdere', det betød at arbejdet på IBM ebbede ud - næsten. Ind imellem kom der dog lidt småopgaver - og stadig vedligeholdelse af noget tidligere udviklet. Nej, jeg har bestemt *aldrig* bygget fejl ind i mine programmer med vilje - for senere at kunne tjene penge på at reparere dem, dertil har jeg for megen faglig stolthed.

Softwarehuset's VM kunde, besluttede at holde op med at køre VM, så jeg blev i en periode 'konverteret' til at programmere i PC-DOS REXX. Ikke noget godt bytte, men noget af det, jeg havde udviklet til VM/CMS, *kunne* faktisk godt køre på en DOS PC, med REXX. Det var dog kun en kort periode, så jeg begyndte at se i øjnene, at jeg omsider var ved at blive til en rigtig pensionist!

Så en dag ringede CMSC. Min afløser havde fået barselsorlov, og man havde mægtig travlt, så man ville gerne se mig igen i CSMC - nogenlunde regelmæssigt. Min afdelingschef har oven i købet nævnt at han gerne ville have mig ansat i et 'andet' firma. ét der kunne give ham adgang til min arbejdskraft på lidt bedre betingelser. Det blev dog ikke til noget, men alt i alt gik der vel 3 - år før jeg helt holdt op med at arbejde for IBM.

Min 'gamle' IBM PC, var i mellemtiden blevet udskiftet til en mere moderne og langt stærkere clone, så da Jon spurgte om jeg var interesseret i at 'låne' hans PC/370 kort til min private PC, måtte jeg desværre sige fra. Det pågældende PC/370 kort kan kun køre på en PC med micro channel, og micro channels findes kun i IBM PS/2 og IBM RISC maskiner. Kortet ligger måske stadig og venter på mig. Det gør lidt ondt, fordi med al den fornøjelse jeg har haft af at arbejde med VM i min karriere, ville det da være rart (og vistnok også velfortjent) at have sit eget VM-system at hygge sig med i sit otium.

Jeg har, indtil nu, hele tiden haft mulighed for at køre VM hjemmefra, først gennem chatterboxen, men senere, da *den* døde, gennem Lan Distance, som jeg omsider har fundet ud af at installere på min egen PC. Problemet med at installere Lan Distance var helt banalt, men denne gang var det *ikke* en *dårlig* fejlmeddelelse, det var en *manglende* fejlmeddelelse.

Kort tid efter jeg var blevet sendt på pension, kom der en note (altså et elektronisk brev) fra en fyr hos Philips i Holland. Joh, sådan noget kan godt lade sig gøre, fordi det meget berømte og omtalte internet, er forbundet med IBM's VNET et eller andet sted henne i USA. Noten handlede om Jon, han var ved at nærme sig de 50 år, og man havde fået den strålende idé at alle hans venner og bekendte fra hele kloden, skulle skrive et fødselsdagsdokument til ham. Det skulle så placeres på Internettet, så alle hans venner, bekendte og kolleger, kunne give ham deres hilsen og hyldest. Jon er virkelig blevet meget populær, dels på grund af hans pipelines,

men også fordi han er en ualmindelig hyggelig fyr. Det var Melinda fra Princeton universitetet, der havde fortalt Philips-manden om mig. Melinda kender mig kun gennem Jon. Hun er i øvrigt en af IBM's meget aktive VM-kunder. Hun har blandt andet skrevet en bog om VM, hvorfor den er der og hvem der har betydet noget for dens udvikling og sådan noget. Da jeg nu er den, der oprindeligt gjorde Jon interesseret i VM, var det helt naturligt at jeg skulle være med til at skrive i fødselsdagsdokumentet. Nu er forfatterskab ikke lige mig. Jeg husker stadig tydeligt hvor inderligt min dansklærer i mellemskolen og jeg hadede hinanden. Og så på engelsk. Jeg fik da skrevet et kapitel, og jeg blev aftvunget et billede, så hvis man har adgang til Internettet, kan man se dokumentet - og et par billeder af/med mig - på:

<http://vm.marist.edu/~piper/>

Det blev et imponerende 'dokument'. Desværre var jeg ikke tilstede ved overrækkelsen, det skete i USA, men rygterne fortæller at Jon blev *meget* glad for det, hvad jeg godt kan forstå. Jeg har selv læst det flere gange, og det indeholder en masse fornøjelige anekdoter og fortællinger fra mennesker fra hele verden - og naturligvis masser af fornøjelige billeder. Mange af bidragyderne har aldrig mødt Jon, de kender ham kun af omtale, eller fordi de har haft glæde af pipelines. Og andre, som f.eks. mig selv, har arbejdet sammen med ham og kendt ham i rigtig mange år.

Når jeg tænker tilbage, må jeg erkende at jeg kan blive helt stakåndet over den kolossale udvikling, som jeg har været så heldig at være en del af.

Den maskine vi kørte VM på, da jeg holdt op med at være systemprogrammør omkring 1982, havde en hukommelse på 8 MB og den kunne udføre ca. 1.1 million instruktioner i sekundet. Min første PC havde 10 MB hukommelse og i følge instruktionsbogen, kunne den udføre ca. 4.5 millioner instruktioner i sekundet. Min nuværende PC har 32MB hukommelse, og den er vel ca. 25 gange så hurtig som den første. Nu ved jeg godt at man ikke skal sammenligne instruktioner, og heller ikke PC'ere med værts systemer, men det er en kendsgerning at min PC kører REXX programmer omkring 3,5 gange hurtigere end IBM's VM system. Nu bør jeg lige fortælle at IBM's VM system har 5 processorer, så der er altså 5, der kan køre samtidig. Konklusionen må dog være at de små personlige computere efterhånden er blevet så stærke at de på mange områder (specielt når det gælder talknusning og grafik) kan sammenlignes med de konventionelle værtssystemer og intet tyder på at udviklingen stopper!!!!

Min tid i IBM har været fyldt med gode oplevelser. Jeg har truffet en masse rare mennesker, både kolleger og kunder. De fleste har jeg haft et vældig godt samarbejde med, og kun et par gange har jeg haft chefer, jeg ikke rigtig brød mig om.

Jeg kan godt blive helt bedrøvet ved tanken om alle de gode kolleger, jeg har mødt, som jeg aldrig får en chance for at se igen. Det gælder både lokale, som jeg har arbejdet tæt sammen med, men også udlændinge, som jeg har mødt på kurser og seminarer.

Jeg har også rejst en del, så jeg har fået set en hel del af verden, som jeg ikke ville have oplevet uden IBM. Selvfølgelig har arbejdet til tider hindret mig i at være så meget turist, som jeg kunne ønske, men arbejdet er dog det vigtigste og lidt turisme har der dog været tid til. Og så alle de gode middage, jeg har spist på IBM's regning på en masse dejlige restauranter . . .

I mange år følte jeg at IBM var et firma, der satte pris på, og respekterede deres medarbejdere. Da det begyndte at gå ned ad bakke for IBM, ændrede denne opfattelse sig desværre radikalt. Det er så meget mere grotesk, når man tager i betragtning at IBM Danmark *aldrig* har haft økonomiske problemer. Jeg har set gode kolleger blive behandlet på en måde der gør, at jeg ikke er sikker på jeg ville begynde en karriere i IBM, sådan som IBM er i dag. Jeg må dog skynde mig at anføre at chefen for CMSC, har kæmpet for at hans afdeling, bevarede noget af den gode gamle IBM ånd. Han har altid *vist* at han bekymrede sig for sine medarbejdere, men han har naturligvis ikke altid fået lov at gøre *helt* som han ville. Jeg har dog haft en fornemmelse af at CMSC var mere som det IBM, jeg holdt af, end mange andre IBM afdelinger.

Tilbage har jeg nu min familie, min private PC, der kørte OS/2 (så længe der vedblev at være liv i OS/2) og som har KEDIT, men som alle andre har jeg været tvunget til at skifte til Windows, og dermed er *alt* det sjove ved PC'en forsvundet, jeg tror *aldrig* jeg vil tilgive IBM at de droppede OS/2. En formildende omstændighed er at Windows kan køre REXX, og KEDIT til Windows er langt bedre end *den* til OS/2.

Jeg har nu meget mere fritid, end jeg har været vant til, jeg fik en blid overgang til mit otium, fordi jeg havde lidt løsarbejde på IBM i begyndelsen. Jeg har en masse gode minder - som jeg her har forsøgt at fortælle om, så jeg må i sandhed sige at jeg *ikke* er at beklage.

This page intentionally left blank.

Ordforklaring

	1401	1401 var den første kommercielle EDB maskine fra IBM, der fik succes.
	1620	1620 var en lille primitiv matematik maskine. Da den var billig, havde 'man' råd til at lege med den.
	/360, /370 ...	En ny maskin arkitektur, der stadig benyttes (dog kraftigt udvidet) i IBM's værts-maskiner
A	APL	En matematisk notation der er blevet omsat til et computer program
	Appikations program	Et program der er skrevet til at udføre en ganske bestemt opgave
	Assignment	Betegner en udstationering til et andet IBM selskab. Det vare som regel to eller tre år.
	Autocoder	en oversætter til 1401 kildeprogrammer
B	Binære tal	Tal i to-tal systemet. D.v.s. tal består udelukkende af 0 og 1, således skrives 25 f.eks. 11001.
	BIT	Egentlig et tal i to-tal-systemet. En bit kan have værdien nul eller én.
	BOS	B asic O perating S ystem. Det mindste og mest primitive operativ system til System /360
	BPS	B it p r. S ekund. Et mål for transmissions hastigheden på en telefon-linie
	BYTE	Den mindste adresserbare enhed i en System /360 maskine. En byte er 8 bits.
	BWS	B ig W hite S witch. Den knap man skal trykke på, for at slukke en PC. Det er evt. <i>det</i> man gør, når alt andet ikke kan hjælpe.
C	CAD	C omputer A ided D esign, en metode til at lave tekniske tegninger på en computer
	CAM	C omputer A ided M anufacturing. Out fra en CAD tegning kan direkte bruges til at dirigere en numerisk styret værktøjsmaskine
	CMS	C onversational M onitor S ystem, et operativsystem, der kun kan køre i en virtuel maskine under VM. Dette system gør livet let for terminalbrugere. CMS betød oprindeligt Cambridge Monitor System.
	CMSC	C ompetitive M arketig S upport C enter, den IBM afdeling, hvor jeg havde fornøjelsen at tilbringe mine sidste år i IBM.
	COBOL	C ommon B usiness O riented L anguage, et programmeringssprog beregnet til kommercielle programmer.

D	DASK	D ansk A ritmetisk S ekvens K alkulator. Danmark første EDB maskine. Den blev bygget af de folk, der senere blev til RegneCentralen.
	DEVICE	Det vi kaldte alle de apparater man kan hægte på en computer
	DHI	D ansk H ydraulisk I nstitut
	DOS	D isk O perating S ystem. Dette mellemste System /370 operativ system. Disk-baseret. Næsten samme system som TOS. DOS hedder nu om dage VSE, og DOS er om dage det oprindelige styresystem til en PC
	DP	D ata P rocessing. Forkortelse, der normalt henviser til salgsafdelingen for Data Maskiner
	DTU	D anmarks T ekniske U niversitet, tidligere Danmarks Tekniske Højskole, og endnu tidligere Polyteknisk Læreanstalt
E	EDB	E lektronisk D ata B ehandling
	EDP	E lectronic D ata P rocessing
	EDITOR	Sædvanligvis et program, der bruges til at oprette og vedligeholde tekstbaserede filer. Næsten som en skrivemaskine. Benyttes flittigt ved programmering
	ESA	E nterprise S ystems A rchitecture, en væsentlig udbygning af /360 og /370 arkitekturen.
	EXEC & EXEC2	Oprindelig samling af kommandoer, der kan udføres som én kommando. Både EXEC og især EXEC2 kan anvendes til programmering
F	FORTRAN	F ORMula T RANslation det første high level programmerings sprog - beregnet til matematiske programmer, altså det første sprog man kunne bruge uden at skulle kende maskininstruktionerne.
H	HASP	H uston A utomatic S pooling and P riority S ystem, et SPOOL system til de forskellige OS'er
	Hexadecimale tal	Tal i 16-tals systemet. Talrækken skrives med tallene fra 0 - 9 og bogstaver A -F, således at F repræsenterer værdien 15. Af praktiske grunde skrives et hexadecimalt tal ofte som to tal svarende til en byte.
	HostSystem	På dansk: værtssystem. En 'stor' computer, der kan betjene flere brugere samtidig
	HPO	H igh P erformance O ption, en overbygning til

		VM/SP, der gjorde at store maskiner blev bedre udnyttet
I	IBM	I nternational B usiness M achines. Firma'et hvor det hele skete.
	IPL	I nitial P rogram L oad. Den proces, der gør det muligt at læse et kontrolprogram ind i maskinen. Man skal altid have en device med et IPL program, der kan arbejde sammen med maskinens start procedure.
K	K	Egentlig kilo - for tusinde. I EDB verdenen er K dog ofte det samme som 1024 - af hensyn til det binære talsystem
	Kamikaze	Guddommelig vind. Betegnelsen for de Japanske selvmordsbombere under anden verdenskrig, og flyveren fra New York til Poughkeepsie.
	KB	se Kilo Bytes .
	Kilo Byte	Kilo betyder 1000, men kilobyte er som regel 1024 bytes
L	LAN	L ocal A rea N etwork. En måde at få PC'ere til at arbejde sammen. Men placerer diske og printere på servere, således at alle på LAN'en kan benytte dem.
M	MAKRO	En instruktion der udvikler sig til en række instruktioner. Normalt standardrutiner.
	Mega	Mega betyder 1000.000, men i EDB verdenen er M ofte 1.048.576 af hensyn til det binære talsystem (altså $1.024 * 1.024$)
	MB	Se MegaByte
	MegaByte	Mega er en million, men en megabyte er normalt 1.048.576
	MIPS	Et performance mål. Står for M illioner I nstruktioner p r. S ekund. Da det ikke siger meget om maskinen, tolkes det dog ofte som: M isleading I ndicator for P rocessor S peed
	Modem	Et apparat, der omsætter computerens digitale/binære signal. til et analogt signal. det kan sendes på en telefonlinie. Der skal være en tilsvarende boks i den anden ende, for at lave signalet om, så den modtagende computer kan forstå det.
	MVS	M ultiple V irtual S torages. IBM's største operativsystem indtil OS/390 blev annonceret.

N	NODE	Et knudepunkt i et netværk
	NEUCC	N orthern E urope U niversity C omputing C enter, et center, der blev oprettet på Danmarks Tekniske Universitet ved en donation fra IBM.
o	OS/2	Et PC styresystem. Udviklet parallelt med WINDOWS
	OS/MFT	OS med M ultiprogramming with a F ixed number of T asks. Dette er det mellemste af System /360's 'fulde' operativ systemer.
	OS/MVT	OS med M ultiprogramming with a V ariabel number of T asks. Dette er det mest avancerede af System /360's operativsystemer.
	OS/PCP	OS med P rimary C ontrol P rogram. Dette er det mindste af System /360's 'fulde' operativ systemer
	Overlay	En teknik, hvor man kodede programmerne i dele, der ikke alle nødvendigvis skulle være i maskinen samtidig. En meget krævende form for programmering
P	Paritetsbit	En bit systemet automatisk kan tilføje for at sikre at alle karakterer har samme lige eller ulige antal bits. Kan evt. afhjælpe bit-fejl.
	PC	P ersonlig C omputer
	Pipelines	En meget væsentlig programmerings facilitet til CMS og MVS, men desværre ikke til OS/2. Udviklet i DK.
	PSW	P rogram S tatus O rd, et register i en /360 maskine, der peger på den instruktion, man er ved at udføre.
Q	QCC	Q uater C entury C lub. En eksklusiv klub af IBM'ere, der har været ansat i mere end 25 år.
R	Residency	En kortvarig udstationering til et andet IBM selskab. Et residency varer normalt fire til otte uger, og man forventes at løse en ganske speciel opgave.
	REXX	R estructured E Xtended e Xecutor. Et populært og hurtigt programmeringssprog. Kommer oprindeligt fra VM. Skulle have heddet REX, men dette navn var optaget
	REXXCPS	Et populært mål for hvor hurtigt REXX køre på en maskine. Tallet varierer fra godt tusinde til flere hundrede tusinde

	RISC	Reduced Instruction Set Computer. En type maskine med færre men til gengæld meget hurtige instruktioner
	RJE	Remote Job Entry. Det at man kan sende et job til en computer fra et fjernt sted, ved hjælp af telefonlinier, og evt. få resultatet retur
	RSCS	Remote Spooling Communication Subsystem, et specielt SPOOL system, der giver VM systemer mulighed for at udveksle filer med andre systemer.
S	SIE	Start Interpretive Execution, en instruktion i ESA arkitekturen, der er vital for VM, idet den kontrollerer eksekveringen i en virtuel maskine
	Storage Dump	Storage dump eller bare dump, er en udskrift af hele eller dele af maskinens hukommelse. Brugtes især ved fejlfinding.
	SP	System Product, en af VM's mange afarter
	SPOOL	Simultaneous Periferal Operations On Line. SPOOL er en teknik, der går ud på at håndtere de langsomme perifere enheder i selvstændige tasks, således at selve program behandlingen kan ske hurtigere.
	SPS	Symbolic Programming system er en oversætter til 1620 kildeprogrammer
T	TOS	Tape Operating Sytem. Dette mellemste System /370 operativ system. tape baseret. Næsten samme system som DOS
	TSO	Time Sharing Option, MVS's måde at skabe et brugervenligt miljø for terminalbrugere
	TWA	Trans World Airlines, et stort amerikansk luftfartsselskab
U	UNIX	Et 'andet' operativ system, det kan køre på RISC-maskiner, værtssystemer eller PC.
V	Virtuel/Virtual	Betegner noget der synes at være der, men som ikke er det.
	VM	Virtual Machines. Et styresystem, der giver <i>alle</i> brugere en illusion af a have en maskine for sig selv Oprindeligt var det VM/370. Senere kom VM/SP, VM/XA, VM/ESA
	VSE	Virtual Storage Extended, en udbygning af den trofaste DOS fra starten af /360 arkitekturen.
	VSPC	Virtual Storage Personal Computing, et forlængst

	udgået produkt, der skulle give end-users mulighed for at udføre programmer
WYSIWYG	What You See Is What You Get. Den form for tekstbehandling, hvor man på skærmen ser et billede af dokumentet nøjagtigt som det bliver udskrevet.
Værtssystem	En 'stor' centralt placeret computer, der kan betjene flere brugere samtidig.

X	XEDIT	En texteditor der blev skrevet med der formål at kunne alt hvad de andre texteditorer kunne og så lidt mere
----------	--------------	---

Indhold

Hvordan det begyndte	Side - 1
Min første tid i IBM	Side - 5
1620	Side - 13
System /360	Side - 27
HASP	Side - 39
VM/370	Side - 47
VM i marketing	Side -75
OS/2 og LAN	Side - 107
Bagefter	Side - 117
Ordforklaring	Side - 123